AD NUMBER
AD892611
NEW LIMITATION CHANGE
TO **Approved for public release, distribution unlimited**
FROM **Distribution authorized to U.S. Gov't. agencies only; Administrative/Operational Use; Dec 1971. Other requests shall be referred to Air Force Armament Lab., Eglin AFB, FL.**
AUTHORITY
AFATL ltr, 24 Jun 1974

AFATL-TR-71-169
VOLUME I

CLOSE AIR SUPPORT MISSILE GUIDANCE AND CONTROL STUDY

VOLUME I SIX-DEGREE-OF-FREEDOM SIMULATION

DEPARTMENT OF MECHANICAL ENGINEERING
THE UNIVERSITY OF FLORIDA

TECHNICAL REPORT AFATL-TR-71-169, VOLUME I

DECEMBER 1971

Test & Evaluation

AIR FORCE ARMAMENT LABORATORY

AIR FORCE SYSTEMS COMMAND • UNITED STATES AIR FORCE

EGLIN AIR FORCE BASE, FLORIDA

Close Air Support Missile
Guidance And Control Study

Volume I. Six-Degree-Of-Freedom Simulation

J. Mahig

Test & Evaluation

Distribution limited to U. S. Government agencies only; ~~this report documents the close air support missile guidance and control study;~~ distribution limitation applied December 1971. Other requests for this document must be referred to the Air Force Armament Laboratory (DLWG), Eglin Air Force Base, Florida 32542.

FOREWORD

This report was prepared by the Industrial and Experiment Station, Department of Mechanical Engineering, University of Florida, Gainesville, Florida, under Contract No. F08635-71-C-0073 with the Air Force Armament Laboratory, Eglin Air Force Base, Florida, during the period from 9 December 1970 to 9 December 1971. Lieutenant Robert J. Karner (DLWG) monitored the project for the Armament Laboratory.

The principal investigator for the contractor was Dr. J. Mahig.

This report consists of two volumes. Volume I is devoted to the Six-Degree-of-Freedom Simulation while Volume II is concerned with the Three-Degree-of-Freedom Simulation. This is Volume I.

This technical report has been reviewed and is approved.

Heyward H. Strong

HEYWARD H. STRONG
Acting Chief, Air-to-Surface Guided Weapons Div.

ABSTRACT

This report describes a six-degree-of-freedom program which can be used to determine the trajectory and miss distance of a missile system. The options for the program are such as to permit variation of the aerodynamics, seeker, autopilot, actuator, and missile motor performance for the purpose of accurately simulating a given missile design and evaluating the effects of any changes in system parameters. Sufficient detail has been included in the text in order to minimize the users' effort needed to know how to update or modify the program for his purposes.

TEST + EVALUATION

Distribution limited to U. S. Government agencies only; ~~this report documents the close air support missile guidance and control study;~~ distribution limitation applied December 1971. Other requests for this document must be referred to the Air Force Armament Laboratory (DLWG), Eglin Air Force Base, Florida 32542.

TABLE OF CONTENTS

TABLE OF CONTENTS (CONCLUDED)

LIST OF FIGURES

LIST OF FIGURES (CONCLUDED)

LIST OF TABLES

SECTION I

INTRODUCTION

The purpose of this report is to provide a reference which will enable ready access to the use of a six-degree-of-freedom program which is capable of accurately determining the trajectory and miss distance of a semi-active or passive guided missile. The program is divided into convenient blocks, called modules or subroutines, which do specific tasks: e.g., determine aerodynamic forces, seeker output, state of autopilot, current value of thrust, etc. As a result, the user will be able to easily locate the section of the program where specific calculations are performed and modify them or, if necessary, to add modules to achieve other purposes.

This program has been derived from a program in the library of the North American Rockwell Company, Columbus Division, and is described in NR 70H-232-1 and -2. The purpose of the program was to determine trajectory and miss distance of an air-to-air or air-to-surface missile. This manual goes into somewhat greater detail in identifying the variables and defining coordinate systems than heretofore. This has been possible because of the extensive work carried out with the program by the author in satisfying the requirements of this contract and information supplied by Mr. R. D. Ehrich and P. D. Capcara of the North American Rockwell (NAR) Corporation. The program described below has been modified from the original version supplied to USAF by North American Rockwell Corportion by personnel at the Air Force Armament Laboratory to permit the consideration of the effect of a random spot motion on the miss distance of a laser guided missile. Incorporated into the version presented in this report are additional capabilities which provide an accurate simulation of the quadrant detector, range closure, proportional lead guidance, simplified program reset mechanism for multiple runs, greater target maneuverability in air-to-air simulations, and a more general high frequency actuator routine which will accept either experimental or theoretically derived transfer functions.

SECTION II

PROGRAM DESCRIPTION

2.1 Subroutines, Modules, and Tables

A complete listing of this program appears in Section V. The program consists of three types of subprograms:

(a) Tables of aerodynamic coefficients in block data form.
(b) Modules describing missile subsystems.
(c) Executive subroutines and the main program.

The block data subroutines must be physically located at the front of the program deck after the main program for proper operation. Data is extracted from these tables in the module A1 which makes use of the table look-up subroutines TABL1, TABL2, and TABL3 which form a part of the executive routines.

For each module (e.g., A1, C4) the programmer has the option of using an associated initialization module (e.g., C4I). These initialization modules may be used to compute initial conditions from input data or add to the list of state variables to be integrated. The initialization modules are executed only once at the start of each simulated mission. It is in the modules themselves (e.g., C4) that the derivatives of the state variables are computed. Time is incremented by a fixed amount (Δt) after every other pass since a predictor-corrector integration algorithm is used.

A large block common array, called C, allows the communication of certain variables between modules and subroutines for input/output, integration, and control purposes.

The mathematical relationship of various modules and subroutines are shown in Figure 1, and a corresponding list of the modules is given in Table I.

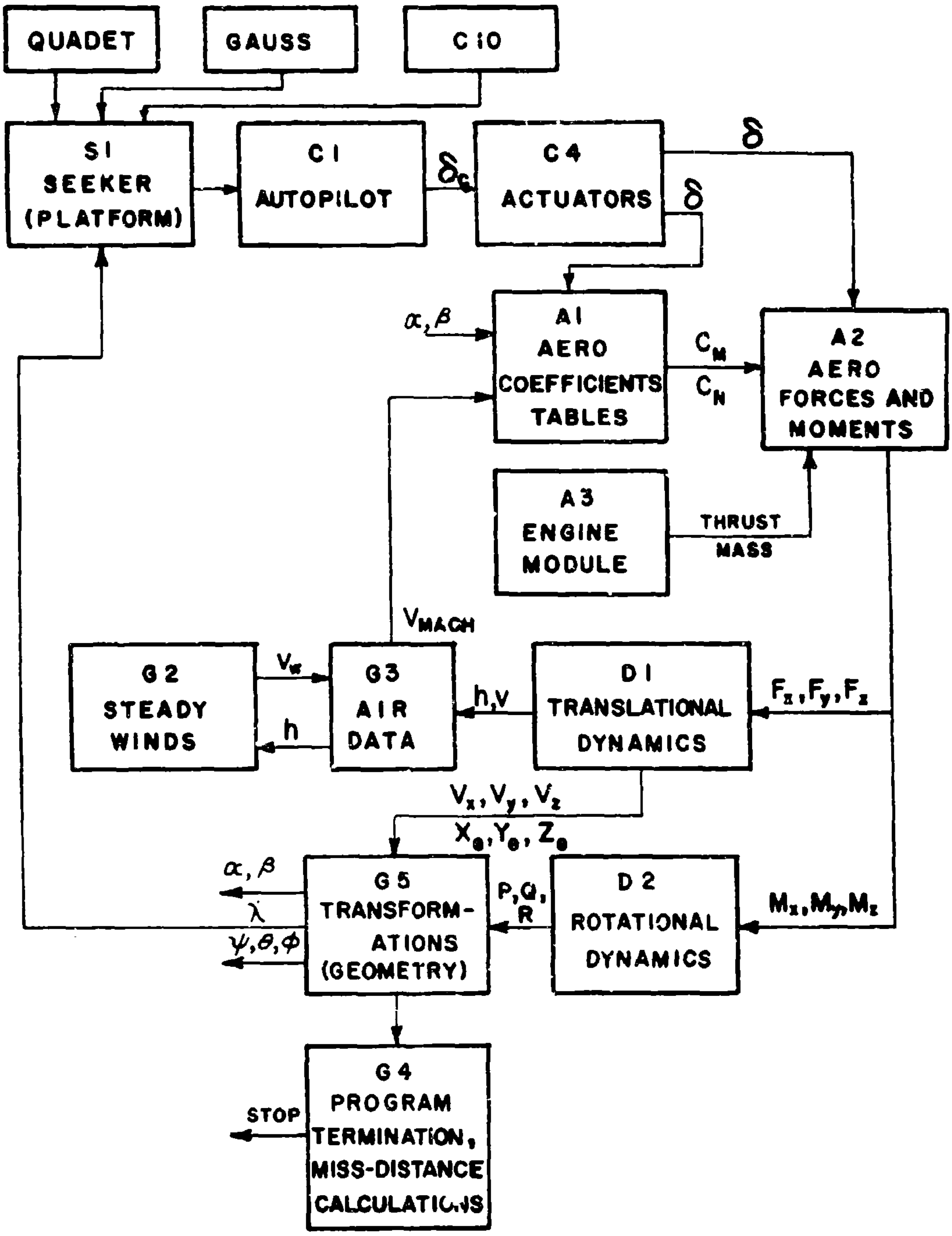


Figure 1. Flow Chart for State Variable Calculations

TABLE I. SUBROUTINE AND MODULE LIST SIX-DEGREE-OF-FREEDOM DIGITAL PROGRAM

I	GEOPHYSICAL AND EXTERNAL ENVIRONMENT
	G2 - Steady winds G3 - Air data - including dynamic pressure, density, speed of sound G4 - Terminal geometry - computes miss distance G5 - Transformations of position and velocity between various coordinate systems
II	SENSORS
	C10 - Spot motion - including boresight error, aiming error, hotspot motion, etc. S1 - Seeker - Seeker performance and platform motion QUADET - Quadrant detector simulator
III	COMPUTERS
	C1 - Autopilot - computes steering commands from seeker output C4 - Actuators - includes flap motion and limits
IV	AIRFRAME
	A1 - Aerodynamics coefficients - table look-up A2 - Aerodynamic forces and moments - in wind axis, includes forces and moments on lugs while missile is on rail A3 - Engine - computes thrust forces as well as c.g. shifts and mass changes.
V	DYNAMICS
	D1 - Translation dynamics of missile - accelerations in body axes are transformed into earth coordinates and integrated into velocities and positions. D2 - Rotational dynamics of missile - computes rotational accelerations and velocities referred to missile body axes.

2.2 A2 - Aero Forces and Moments

Figure 2 shows the relationship between the body axis and wind axis coordinate system. In addition, the coordinate directions are shown for the positive direction of the dimensionless aerodynamics' coefficients in both the body axis system and the wind or primed axis system. The body axis system and the wind axis system are related by the following system of equations:

$$\begin{bmatrix} X_B \\ Y_B \\ Z_B \end{bmatrix} = \begin{bmatrix} & & \\ & \phi' & \\ & & \end{bmatrix} \begin{bmatrix} X'_B \\ Y'_B \\ Z'_B \end{bmatrix}$$

The aerodynamic coefficients are functions of the aerodynamic roll angle (ϕ') and angle of attack α'. It can be seen that the angles ϕ' and α' locate the wind vector in much the same way that a magnitude r and angle θ locate a vector in polar coordinates. With reference to Figure 2, it is apparent that the plane containing the wind vector is obtained by rotating the X_BZ_B plane through ϕ' about the missile centerline (X_B axis). The wind vector is located in this plane by the angle α' measured from the X_B axis. The angles ϕ' and α' are related to the angle of attack α and sideslip β by the following equations:

$$\text{Cos } \alpha' = \text{Cos}\alpha \text{ Cos}\beta$$

$$\text{Sin } \phi' = \text{Sin } \beta/(1-\text{Cos}^2\alpha \text{ Cos}^2\beta)^{\frac{1}{2}}$$

where if α and β are small, one finds

$$\alpha'^2 = \alpha^2 + \beta^2.$$

Since if α and β are small, α' will similarly be small and it will be found that

$$\beta = \alpha'\text{Sin}\phi'$$

$$\alpha = \alpha'\text{Cos}\phi'$$

$$\tan\phi' = \beta/\alpha$$

$$\alpha' = \sqrt{\alpha^2 + \beta^2}.$$

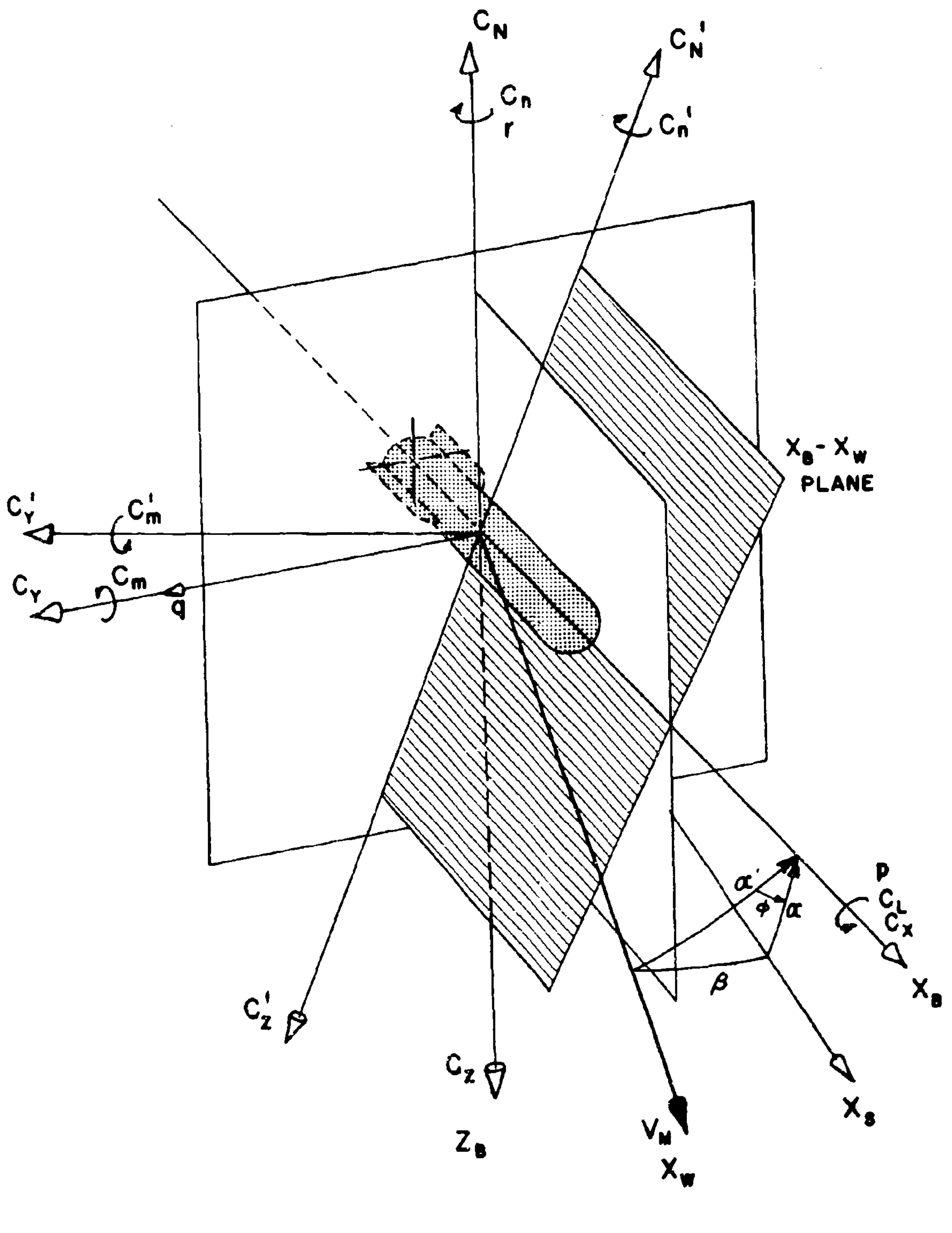


Figure 2. Wind Axis System

ϕ'_A is the aerodynamic roll angle referenced to zero with the missile flying in the + configuration. If the missile is intended to fly in the X configuration, ϕ'_A equals 45° with $\beta = 0$. Thus, $\phi' = \phi'_A - 45°$.

It will be found that the following relationships hold with respect to ϕ and ϕ'_A:

$$\text{Cos } 4\phi'_A = -\text{Cos } 4\phi'$$

$$\text{Sin } 4\phi'_A = -\text{Sin } 4\phi'$$

$$\text{Cos } (\phi_A - 45°) = \text{Cos } \phi'$$

$$\text{Sin } (\phi_A - 45°) = \text{Sin } \phi'.$$

Some of the above relations can be experienced in terms of the angle ϕ'.

In order to facilitate application to the program, Table II lists the correspondence between variable names, commonly used aero symbols, and their COMMON location in the program.

TABLE II. CORRESPONDENCE BETWEEN VARIABLE NAMES, AERO SYMBOLS, AND THEIR COMMON LOCATION

Name	Symbol	Common Location	Name	Symbol	Common Location
CX	C_A	1203	CLDRP	$C_n'/\delta r$	1225
CY	C_Y	1204	CNQ	$C_n'/\delta q$	1226
CZ	C_Z	1205	CLD	$C_\ell'/\delta p$	1227
CLP	$C_{\ell p}$	1206	CLMP	C_m'	1228
CMQ	C_{mq}	1207	CLNP	C_n'	1229
CNR	C_{nr}	1208	BDEFL	$\|\delta\|$	1230
CL	C_ℓ'	1209	CDCM	$C_m'(\phi')$	1231
CM	C_M	1210	BDL	δ_p	1232
CN	C_N	1211	BDM	δ_q	1233
CXO	C_A	1212	BDN	δ_r	1234
CXC	$C_{A'}$(trim)	1213	CDCN	$C_{N'}(\phi')$	1235
CNPT	$C_{N'}(\alpha')$	1214	CL2	$C_{\ell'}(\phi)$	1240
CY2	$C_{Y'}(\phi')$	1215	CL3	$C_{\ell'}(\phi)$ lug	1241
CMO	$C_{M'}(\alpha')$	1217	CNPU	$C_{N'}(\phi',\alpha')$	1244
CN2	$C_{N'}(\phi')$	1218	CYPU	$C_{Y'}(\phi)$	1245
CZQ	$C_{N'}/\delta q$	1219	CMP	$C_{m'}(\alpha',\phi')$	1247
CZR	$C_{N'}/\delta r$	1220	CNP	$C_{n'}(\phi')$	1248
CMDQP	$C_{M'}/\delta q$	1221	CLR	$C_{\ell'}(\phi')$	1249
CMR	$C_{M'}/\delta r$	1222	CZP	$C_{N'}$	1250
CYR	$C_{Y'}/\delta r$	1223	CYP	$C_{Y'}$	1251
CYQ	$C_{Y'}/\delta q$	1224			

2.3.1 D1,D2 - Translational and Rotational Dynamics Module

The following list of symbols applies to the equations of motion which are developed in paragraph 2.3.3 for modules D1 and D2.

2.3.2 List of Symbols

m - F(t) instantaneous mass (slugs)

p - Rolling velocity = angular velocity along X axis (rad/sec)

q - Pitching velocity = angular velocity along Y axis (rad/sec)

r - Yawing velocity = angular velocity along Z axis (rad/sec)

I_X - Moment of inertia about X axis (slug-ft)

I_Y - Moment of inertia about Y axis (slug-ft)

I_Z - Moment of inertia about Z axis (slug-ft) ($I_Z = I_Y$ for a perfectly symmetrical missile)

U - Linear velocity along the X body (X_E) axis (ft/sec)

$V = U_\alpha$ - Linear velocity along the Y body (Y_E) axis (ft/sec)

$W = U_\beta$ - Linear velocity along the Z body (Z_B) axis (ft/sec)

X_B, Y_B, Z_B - Airframe axis system that moves with airframes

X_e, Y_e, Z_e - Earth coordinates

α - Angle of attack = angle between a fuselage reference line and the relative wind in the X_B, Z_B plane (rad)

$$\text{Tan}\ \alpha = W/U; \alpha = W/U$$

β - Angle of sideslip (rad)

$$\text{Tan}\ \beta = V/\ U^2 + W^2;\ \beta = V/U$$

Euler angles
ψ, θ, and ϕ - ψ is the rotation about Z_B, θ is the rotation about Y_B, and ϕ is the rotation about X_B in that order (rad)

g - Acceleration due to gravity (ft/sec^2)

T - Thrust along X_B

C_N, C_Y, C_C, C_R, C_m, C_n - Dimensionless aerodynamics coefficients (body axes)

C'_N, C'_Y, C'_C, C'_R, C'_m, C'_n - Dimensionless aerodynamics coefficients (primed axes system - Figure 2)

- Density (slug/ft^3)

q_0 - Dynamic pressure = $\frac{1}{2}\rho U^2$ (lb/ft^2)

S - Body reference cross sectional area (ft^2)

ℓ - Reference body length (ft)

ΔX - Shift of center of gravity from a reference point along the X_B axis (ft) - negative aft

δ - Control surface deflection (rad)

δ_q - Control surface deflection to give pitching motion (rad)

δ_p - Control surface deflection to give rolling motion (rad)

δ_r - Control surface deflection to give yawing motion (rad)

C_{ij} - Dimensionless aerodynamic derivatives

d' - Aerodynamic or wind angles of attack (rad) - Figure 2

ϕ' - Aerodynamic roll angle (rad) - Figure 2

ϕ'_A - Aerodynamic roll angle (rad) referenced to zero when flying in the + configuration. $\phi'_A = \phi' + 45°$

2.3.3 Equations of Motion

The six-degree-of-freedom equations of motion implemented in the computer program in terms of the body axes are given below*. (See Figure 3 for coordinate system orientation.)

Longitudinal Force

$$\Sigma F_X = m[U + U_\alpha q - U_\beta r] = q_0 S_\pi C_C - mg\mathrm{Sin}\theta + T$$

Lateral Force

$$\Sigma F_Y = m[d/dt(U_\beta) + Ur - U_\alpha p] = q_0 S_\pi [\mathrm{Cos}\phi' C'_Y - \mathrm{Sin}\phi' C'_N] + mg\mathrm{Cos}\theta\mathrm{Sin}\phi$$

Vertical Force

$$\Sigma F_Z = m[d/dt(U_\alpha) + U_\beta p - Uq] = -q_0 S_\pi [\mathrm{Cos}\phi' C'_N + \mathrm{Sin}\phi' C'_Y] + mg\mathrm{Cos}\theta\mathrm{Cos}\phi$$

Rolling Moment

$$\Sigma M_X = I_X p = q_0 S_\pi \ell [C_\ell + \ell/2UC_{\ell_p} P]$$

Pitching Moment

$$\Sigma M_Y = I_Y q + (I_X - I_Z)pr = q_0 S_\pi \ell [\mathrm{Cos}\phi' C'_m + \mathrm{Sin}\phi' C'_n + \ell/2U\, C_{mq} q - \Delta X/\ell (\mathrm{Sin}\phi' C'_Y + \mathrm{Cos}\phi' C'_N)]$$

Yawing Moment

$$\Sigma M_Z = I_Z r + (I_Y - I_X)pq = q_0 S_\pi \ell [\mathrm{Cos}\phi' C'_n - \mathrm{Sin}\phi' C'_m + \ell/2U\, C_{n_r} r - \Delta X/\ell (\mathrm{Cos}\phi' C'_Y - \mathrm{Sin}\phi' C'_N)]$$

* $U \sim U$ velocity in X direction
$V \sim U_\alpha$ velocity in Y direction
$W \sim U_\beta$ velocity in Z direction

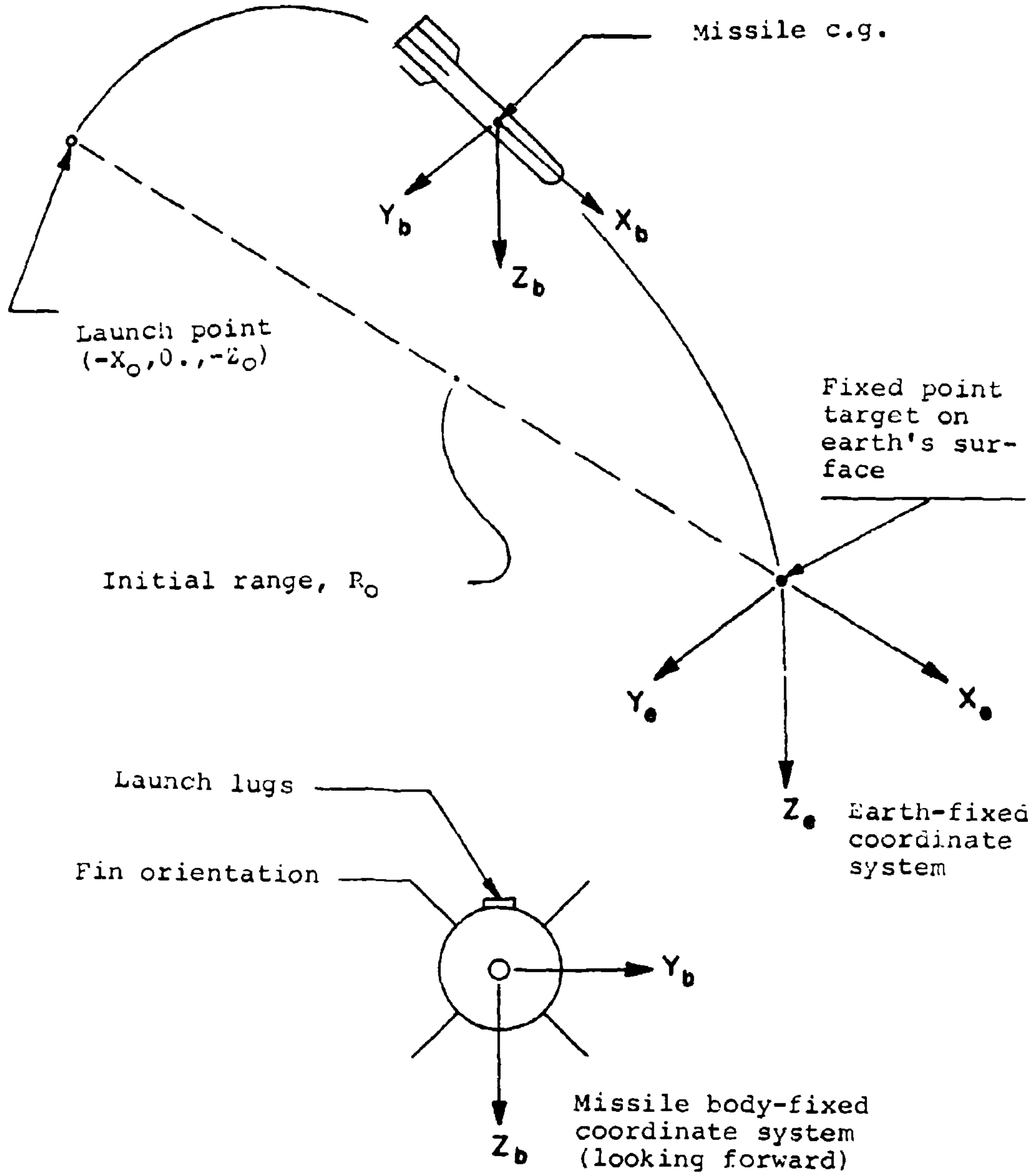


Figure 3. Definitions of Angles and Coordinate Systems

The Euler transformations of

$$\dot{\Psi} = p = r\cos\phi/\cos\theta + q\sin\phi/\cos\theta$$

$$\dot{\theta} = q = q\cos\phi - r\sin\phi$$

$$\dot{\phi} = r = p + (r\cos\phi + q\sin\phi)\tan\theta$$

The velocity, in terms of the earth axes, can be obtained as:

$$\dot{X} = U\cos\theta\cos\Psi + U_\beta(\sin\phi\sin\theta\cos\psi - \cos\phi\sin\psi)$$
$$+ U_\alpha(\cos\phi\sin\theta\cos\psi + \sin\phi\sin\psi)$$

$$\dot{Y} = U\cos\theta\sin\psi + U_\beta(\cos\phi\cos\psi + \sin\psi\sin\theta\sin\phi)$$
$$+ U_\alpha(\cos\phi\sin\theta\sin\psi - \sin\phi\cos\psi)$$

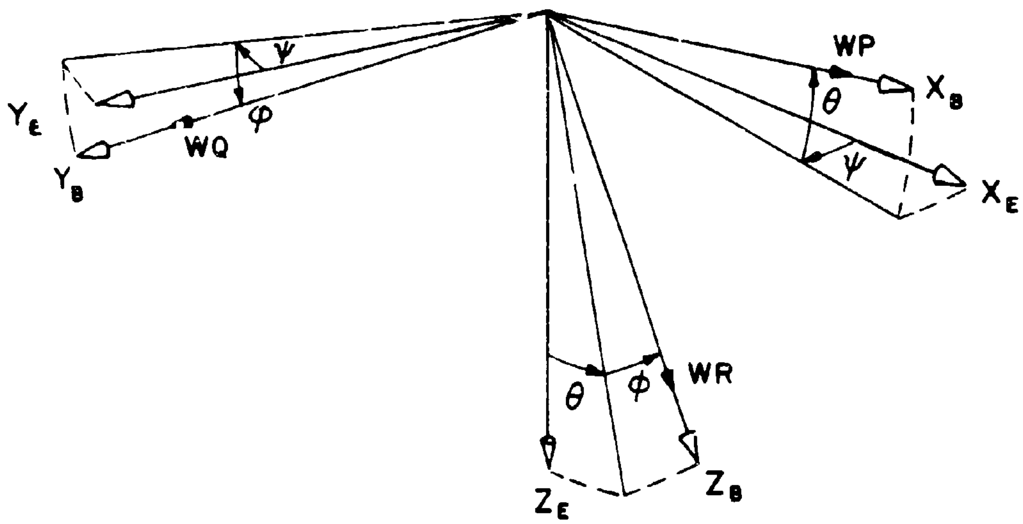
$$\dot{Z} = - U\sin\theta + U_\beta\sin\phi\cos\theta + U_\alpha\cos\theta\cos\phi$$

The Euler angles, shown in Figure 4, and the position of the missile in earth coordinates can be obtained through an integration of the above equations.

The block diagram of the implementation of the equations of motion and the Euler transformations are shown in Figure 5.

2.4 Subroutine G2

This subroutine is called the wind and gust module. This module determines the velocity and direction of the wind. The module assumes that there is no wind above an altitude RHW. Below that altitude the wind direction and magnitude are assumed to be constant throughout a layer RWINC in depth. (It should be noted that RWINC is measured along the line of sight. Since most missiles fly with only small deviation from the original line of sight, the altitude increments, if needed, may be readily estimated.) Two random variables are associated with the wind in each layer: the magnitude and angular orientation which are considered constant in each layer. The mean value of the wind magnitude is VWTE, and its standard deviation is given as SW. The mean value of the angular orientation of the wind in a layer is BPSIW, and the standard deviation of the angular variation is SW1. The current value of the wind magnitude and direction is given by VWTEV and BPSIWV, respectively. The relationship between these mean values and the inertial coordinate system is shown in Figure 6.



Body Coordinate System

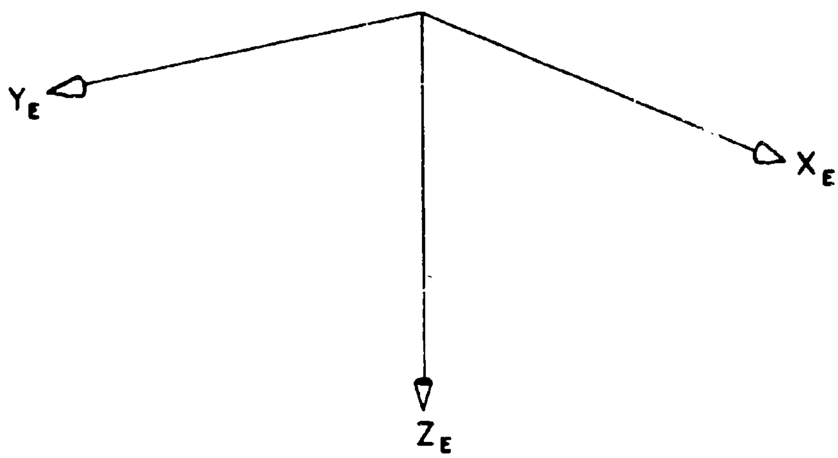


Figure 4. Euler Angles Between Body Axis and Inertial Axis

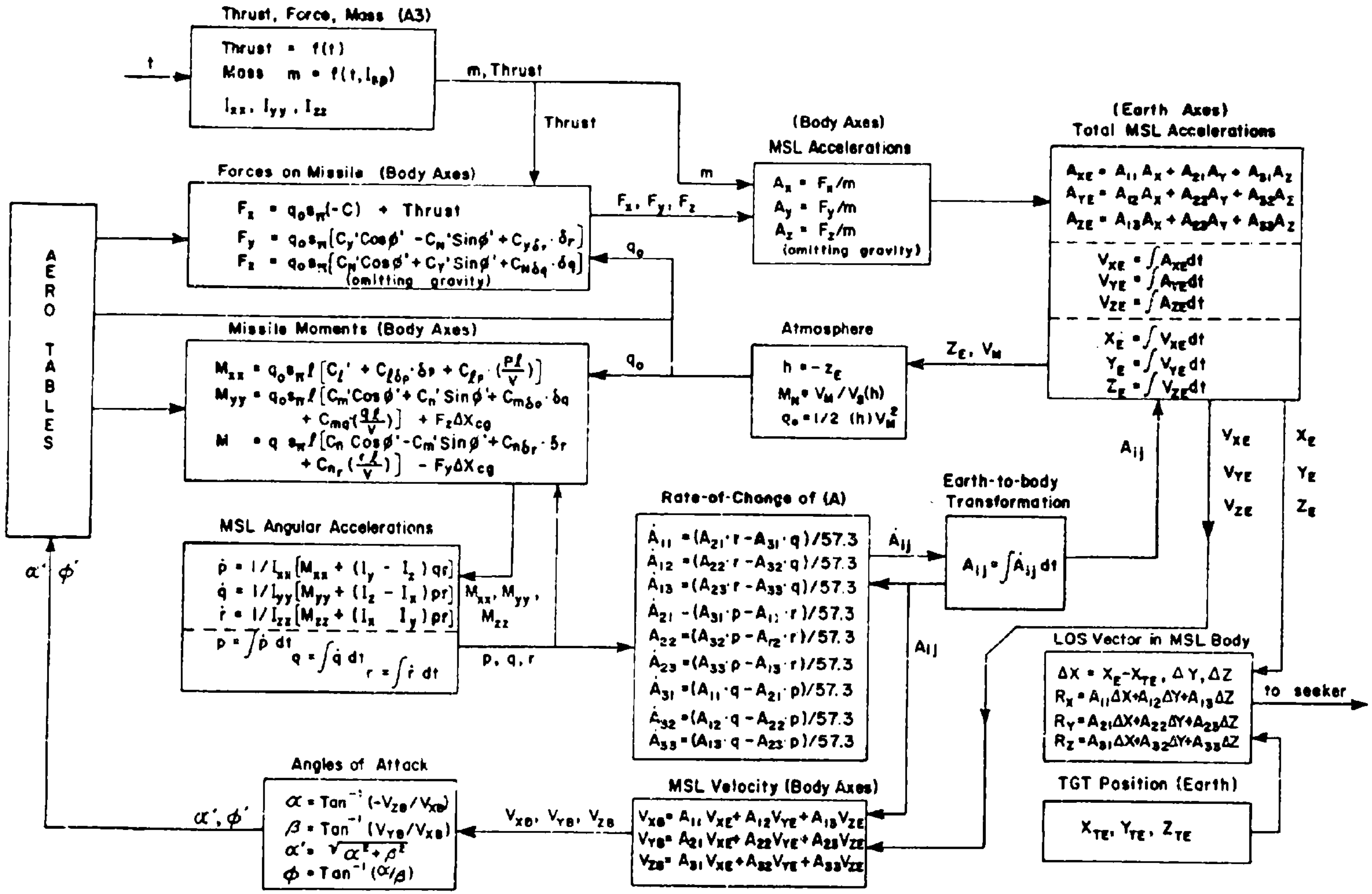


FIGURE 5. SIX-DEGREE-OF-FREEDOM EQUATIONS OF MOTION

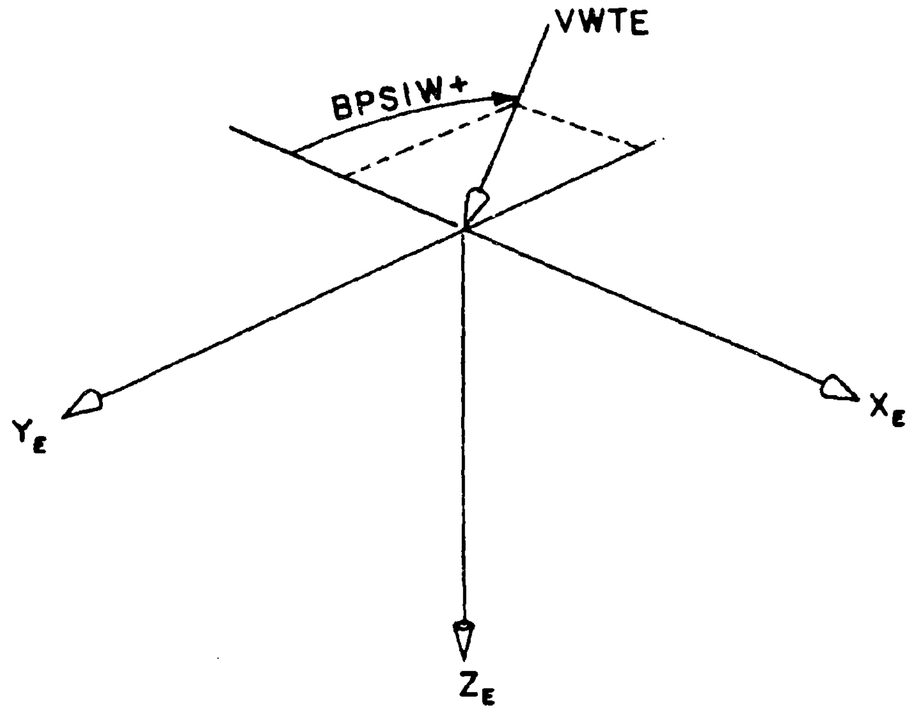


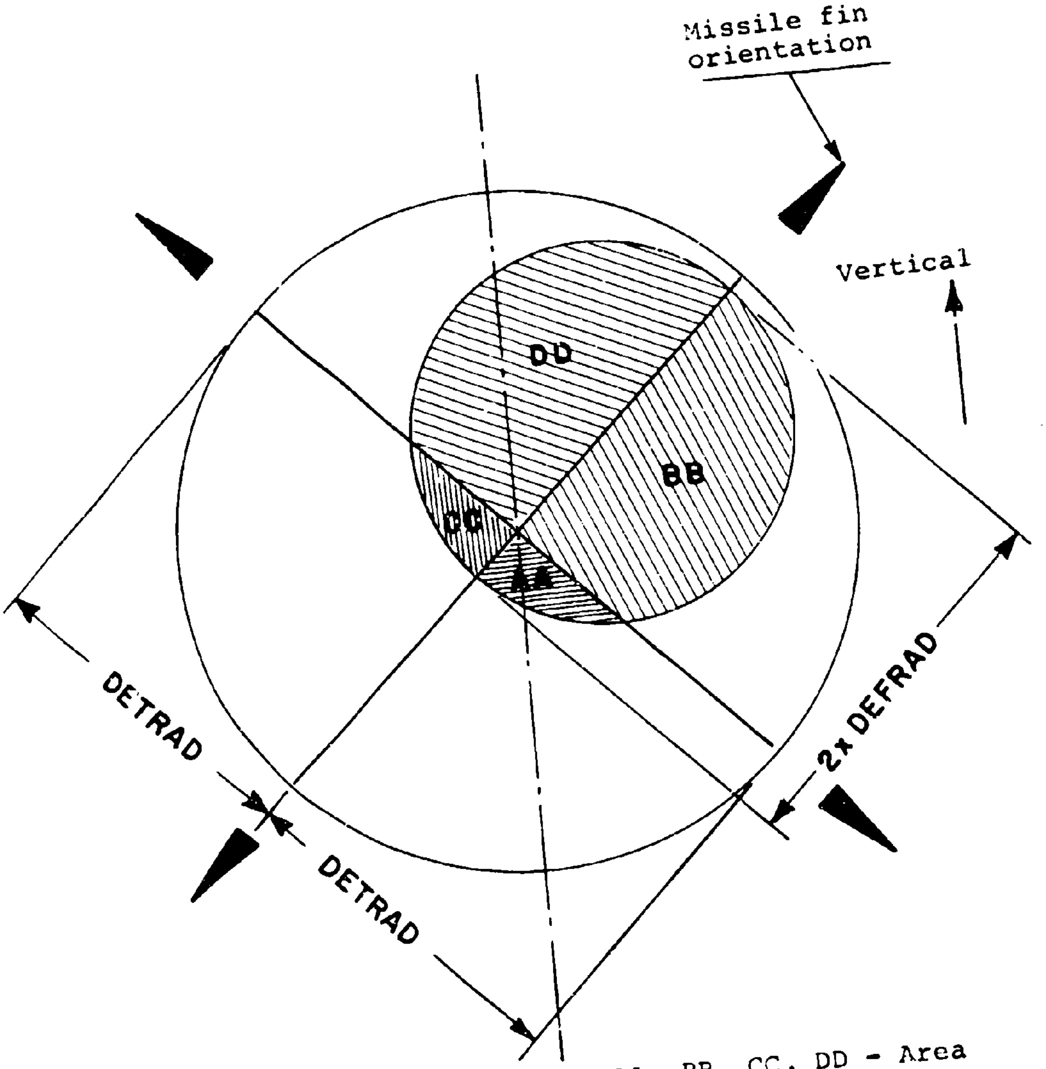
Figure 6. Coordinate System Associated with Wind and Gust Module

2.5 Subroutine C10

This subroutine determines the ground plane coordinates of that point in the area illuminated by the laser beam which the missile seeker physics causes the autopilot to consider the designated target. This distinction is necessary since some seekers are hotspot trackers while others are centroidal trackers. The procedure used to develop this apparent target is accomplished first through the designation of the coordinates of the illuminator (XIL, HILL) which may either be on the forward air controller or on the launch aircraft. The maximum errors generated on the ground are considered to be made up of three parts: the maximum boresight error (BORE), the maximum pointing error (WAND), and the maximum deviation of the hotspot from the resulting beam centroid, which is designated as (RADIUS). Each of these variables is considered a random variable with a uniform distribution. The resulting random variables generated are, respectively, BOREF, WANDF, and SPWID. The variables are considered to vary independently in the XE and YE direction and are then appropriately summed in order to determine the apparent target location. The coordinates of this point are designated as the variables ZLASR and YLASR. The location in earth coordinates may be found by equating ZLASR to XE and YLASR to YE and setting ZE equal to zero.

2.6 Subroutine QUADET

Subroutine QUADET is called by S1 for the determination of the signal generated to the autopilot by the quadrant detector (Figure 7). The quadrant detector is oriented such that the dead zone is in the same direction as the fins, assuming the missile flies in the X configuration. The subroutine determines the current size of the circular image through the assumption that the image size increases inversely proportional to the range of the missile from the laser spot. RT1 is the variable designating the ratio of the size of the current spot to its size at infinity. The laser image on the detector is assumed to be circular. In order to determine the portion of the area of each quadrant covered by the laser image, the area of the detector is subdivided into LT segments. (In the current program LT is set equal to 16.) In order to effect a dead zone, an area round the axis of the coordinate system equal to half the segment width is not included in the area of the image which cover these segments. If a portion of the laser image falls off the assumed circular detector's surface, it is not considered. The variable DETRID is half the instantaneous field of view of the detector in degrees. DEFICS is half the



AA, BB, CC, DD - Area of laser spot in respective quadrant (sq. in.) (less dead zone) shaded area represents defocused spot on quadrant detector

Figure 7. Quadrant Detector Geometry

instantaneous field of view intercepted by the image of the laser spot on the detector. The subroutine will determine if the following blind range and breaklock criteria are met and print this information on the line printer. The breaklock criteria is met if there is no portion of the laser image on any of the four quadrants. The blind range criteria is met if the image of the laser spot on the detector exceeds 90 percent of the total area of the detector.

2.7 Subroutine S1 (Module)

The purpose of the S1 Module is to simulate the response of several types of seeker models and to generate the commands which are transmitted to the autopilot.

The subroutine will simulate the seeker response to either a continuous information source or a sampled data source. This is accomplished by setting the variable OPTKR either to zero or one, respectively. If operating from a continuous information source, the seeker is assumed by the module to be a proportional seeker. In the sampled data mode the seeker can be programmed as either a proportional or a bang-bang seeker by the choice of the magnitude of the variable DEFOCS. If this variable is chosen so that it is equivalent to DELF [detector radius/LT (in current program)], the seeker will simulate a bang-bang laser seeker; whereas, if this variable is chosen so that it is larger than DELF, it will produce a proportional laser processor.

In the sample data mode the seeker will simulate either a pursuit or a proportional navigation system by setting the variable CAGE equal to zero or one, respectively. In the continuous information mode, corresponding changes in the guidance law will occur. In either mode of operation the PLG option may be implemented. This is done by removing the C from the two cards following the PLG OPTION card.

The mode of operation of the subroutine in either mode is to initially determine the true location of the target in the gimbal axis coordinate system (RXG, RYG, and RZG) and then determine the angles the lines of sight make with the RYG, RXG plane and the RZG, RXG planes (BEPSY and BEPSZ, respectively, shown in Figure 8 and Figure 9).

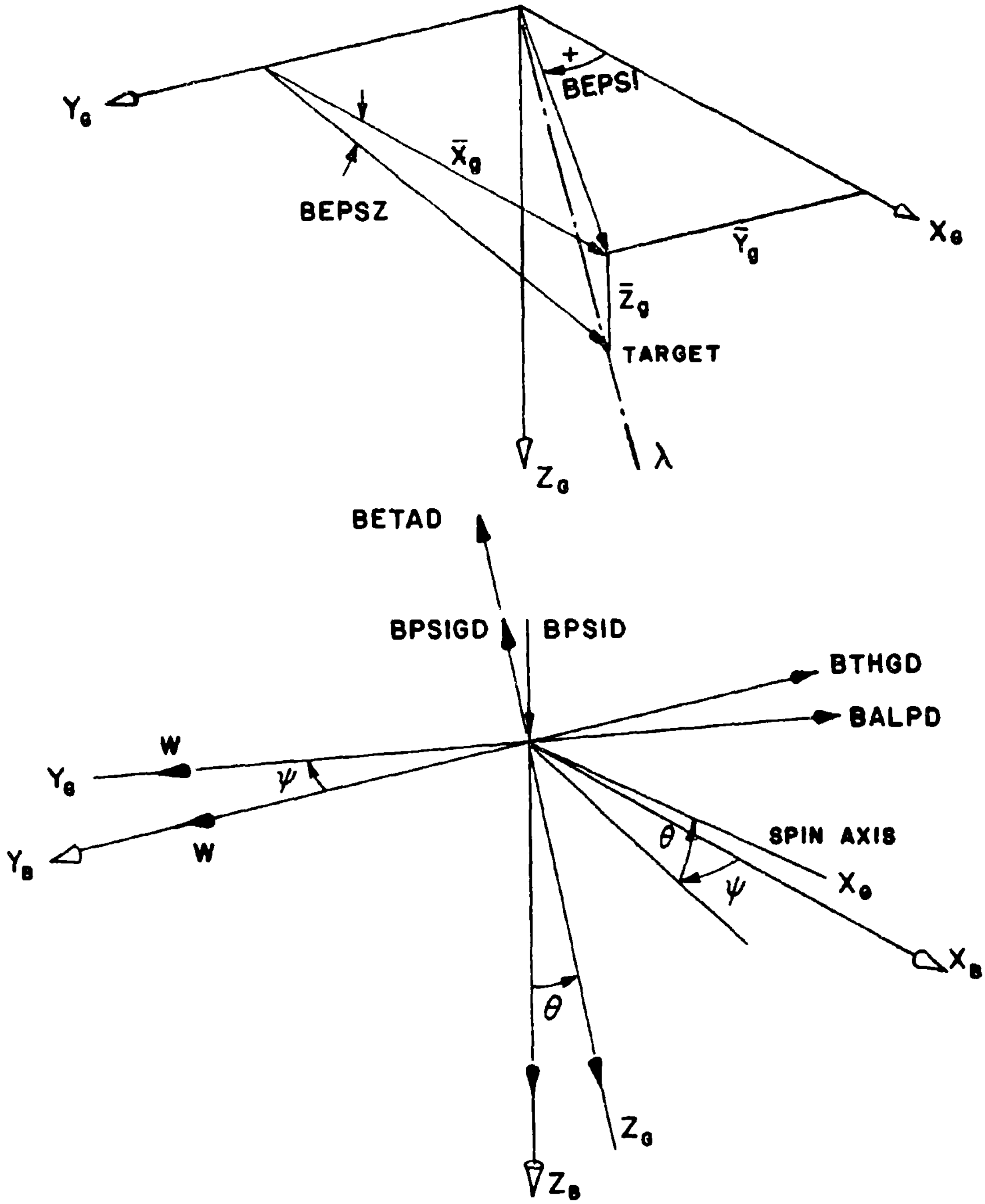
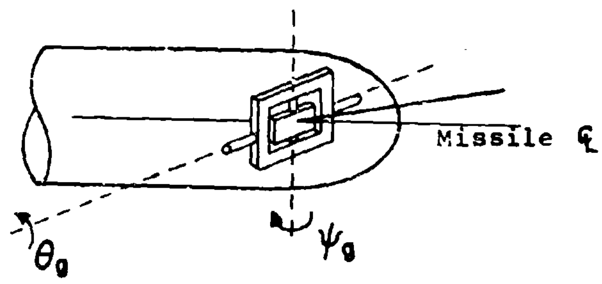


Figure 8. Coordinate Relations Between Body and Gimbal Axis System and the Line of Sight



Gyroscope Platform Gimbal Angles

θ_g outer gimbal - pitch

ψ_g inner gimbal - yaw

Figure 9. Schematic Diagram of Platform Gimbal Angles

The rate of pulse loss is determined by the value, between zero and one, initially given the variable VLAZRP. This is done by comparing a uniformly distributed random variable [C(103)] whose range is also between zero and one with VLAZRP. If it is greater, it is assumed that the information in the pulse is lost. If pulse loss has not occurred, then the apparent location of the target is determined in the gimbal axis coordinate system which has resulted from boresight errors, wander, etc. Subroutine QUADET is then called to determine the output of the quadrant detector. This output is used to generate the required gimbal rate and autopilot commands. If pulse loss has occurred, previously generated commands (e.g., gimbal rate, autopilot signals) are maintained.

In addition, Appendix I shows the mechanics of the coordinate transformation from the body axis to the gimbal axis system for easy reference.

2.8 C1 Autopilot Module

The following high and low frequency autopilot block diagrams are suitable representations for an autopilot that would prove to be consistent with either a proportional or bang-bang seeker. The block diagrams for each of those autopilots are given in Figures 10, 11, 12, and 13. These systems correspond to those mechanized in the program listing found in Section V for the low frequency autopilot and in Appendix III for the high frequency autopilot.

2.9 C4 - Actuator Module

The actuator module simulates the action of the actuator up to a third order system, as shown in Figure 14, which corresponds to a high frequency actuator. Under these conditions it is capable of simulating the dynamics of either a torque balance system whose block diagram is shown in Figure 15, or that of the position loop-controlled actuator shown in Figure 16. It will also simulate the dynamics of an actuator whose transfer function has been determined from hardware test data up to the third order.

The transfer function, given in general form as expressed in this module, is shown below:

$$\frac{\delta}{\delta_c} = \frac{K}{A1 * S^3 + A2 * S^2 + A3 * S + A4}$$

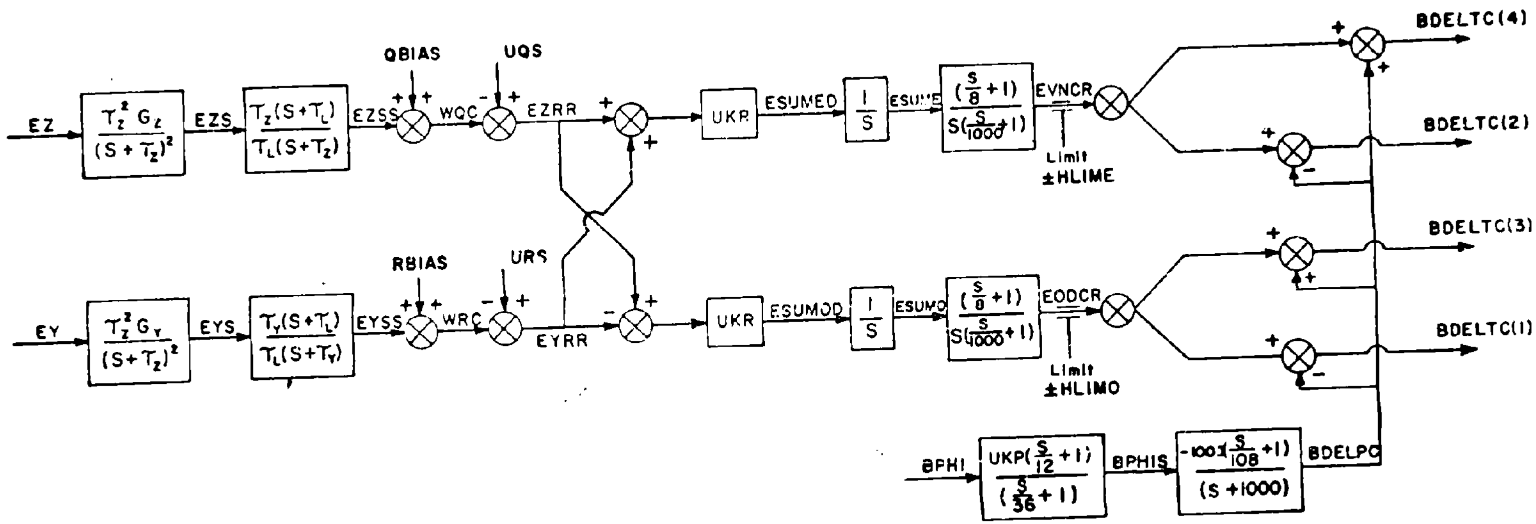


FIGURE 10. AUTOPILOT
HIGH FREQUENCY MODEL

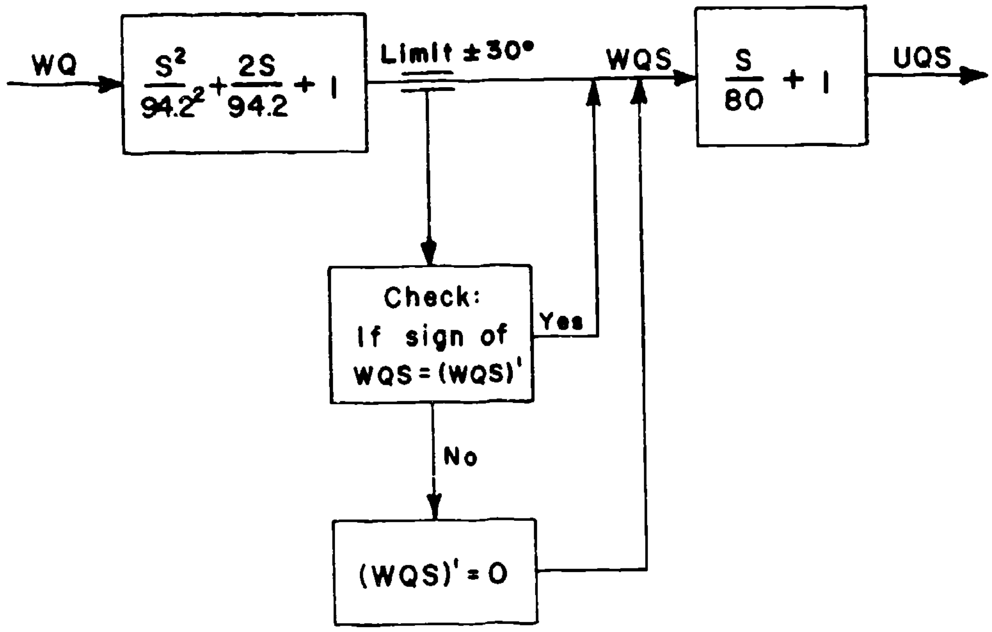


Figure 11. Pitch Rate Gyro

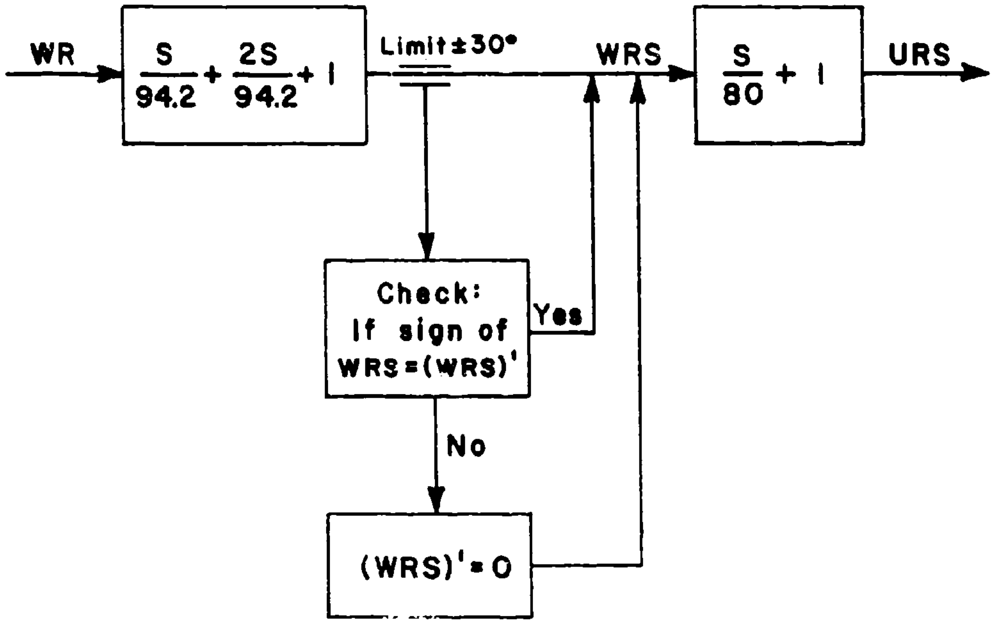


Figure 12. Yaw Rate Gyro

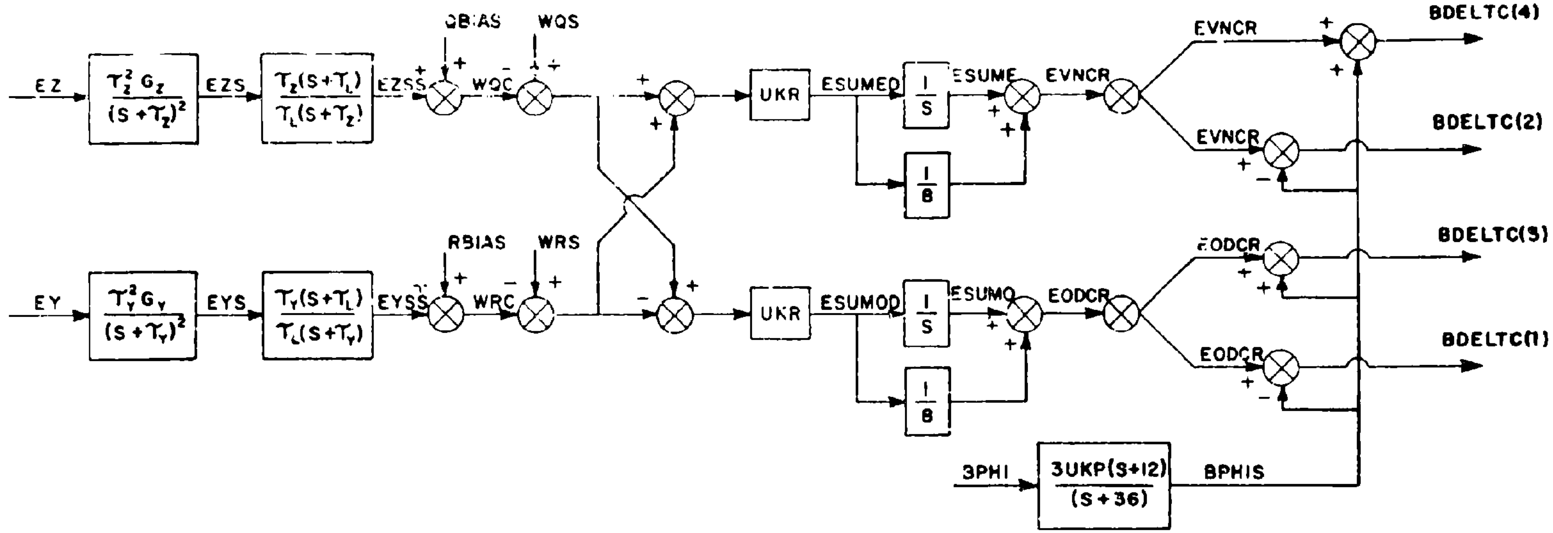


FIGURE 13. AUTOPILOT

LOW FREQUENCY MODEL

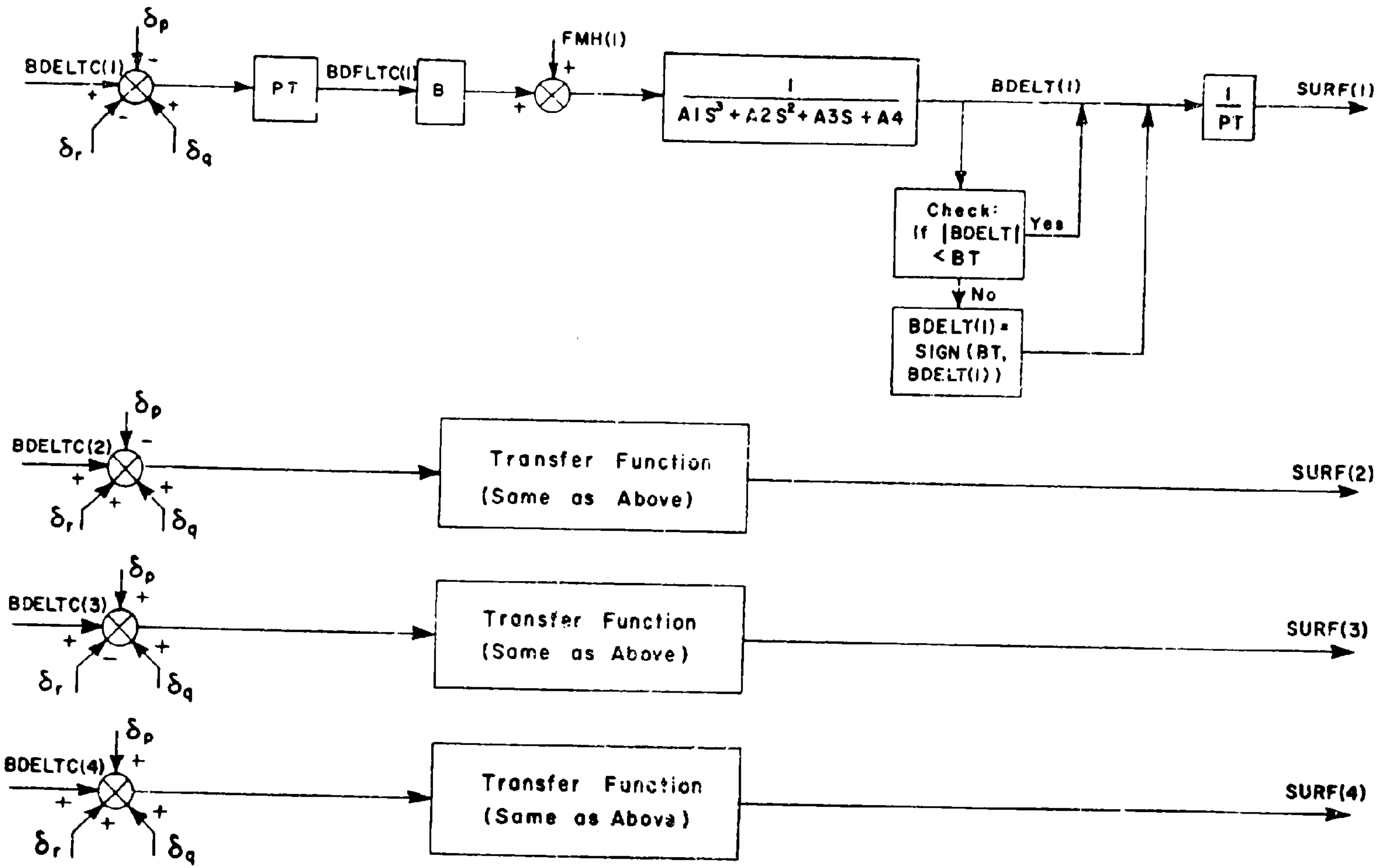


FIGURE 14. HIGH FREQUENCY ACTUATOR

$$\frac{\delta}{\delta_c} = \frac{1}{\left(\frac{S}{16.3}+1\right)\left(\frac{S^2}{180^2}+\frac{.95}{180}S+1\right)}$$

[worst case model from hardware test data]

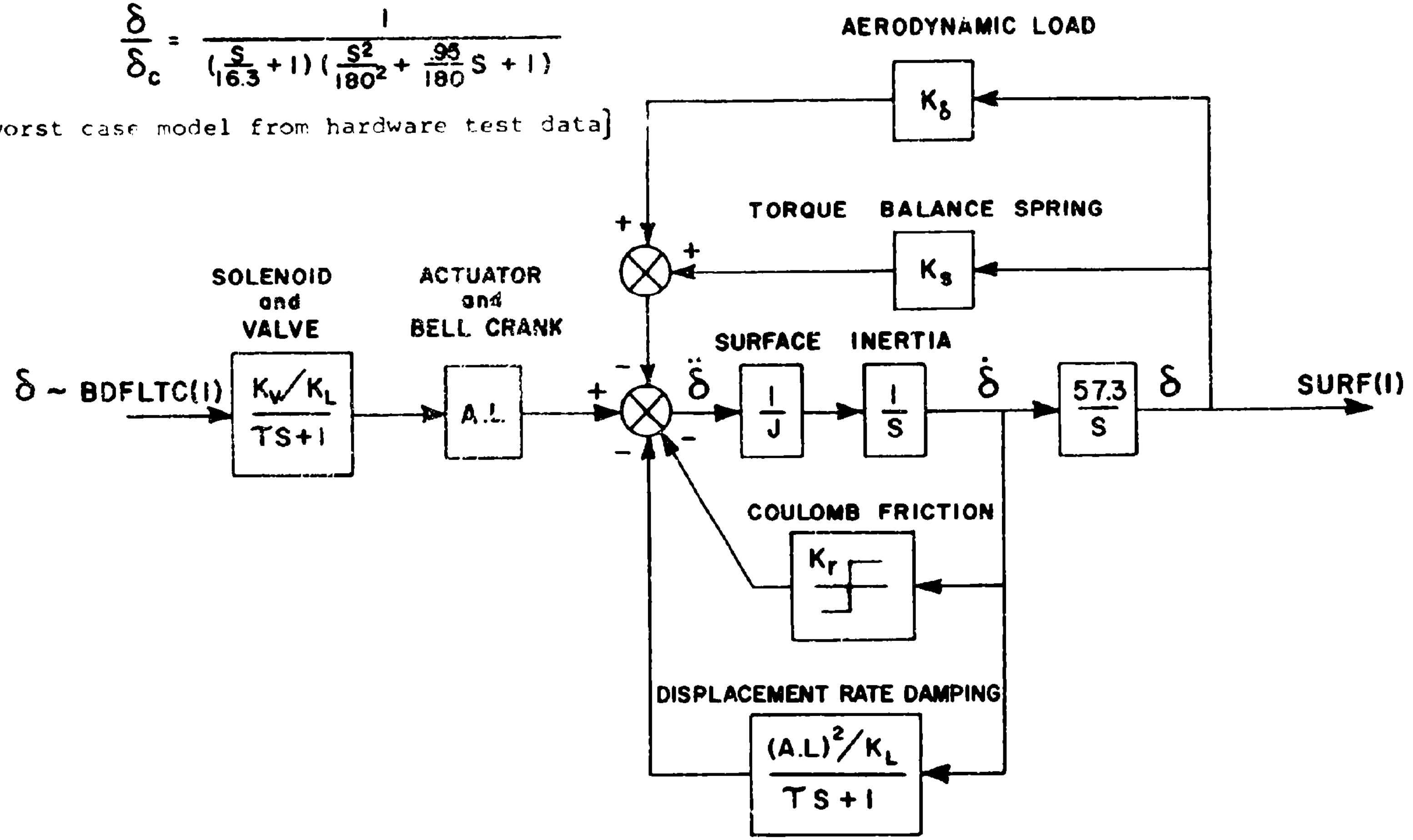


Figure 15. Actuator Torque Balance System

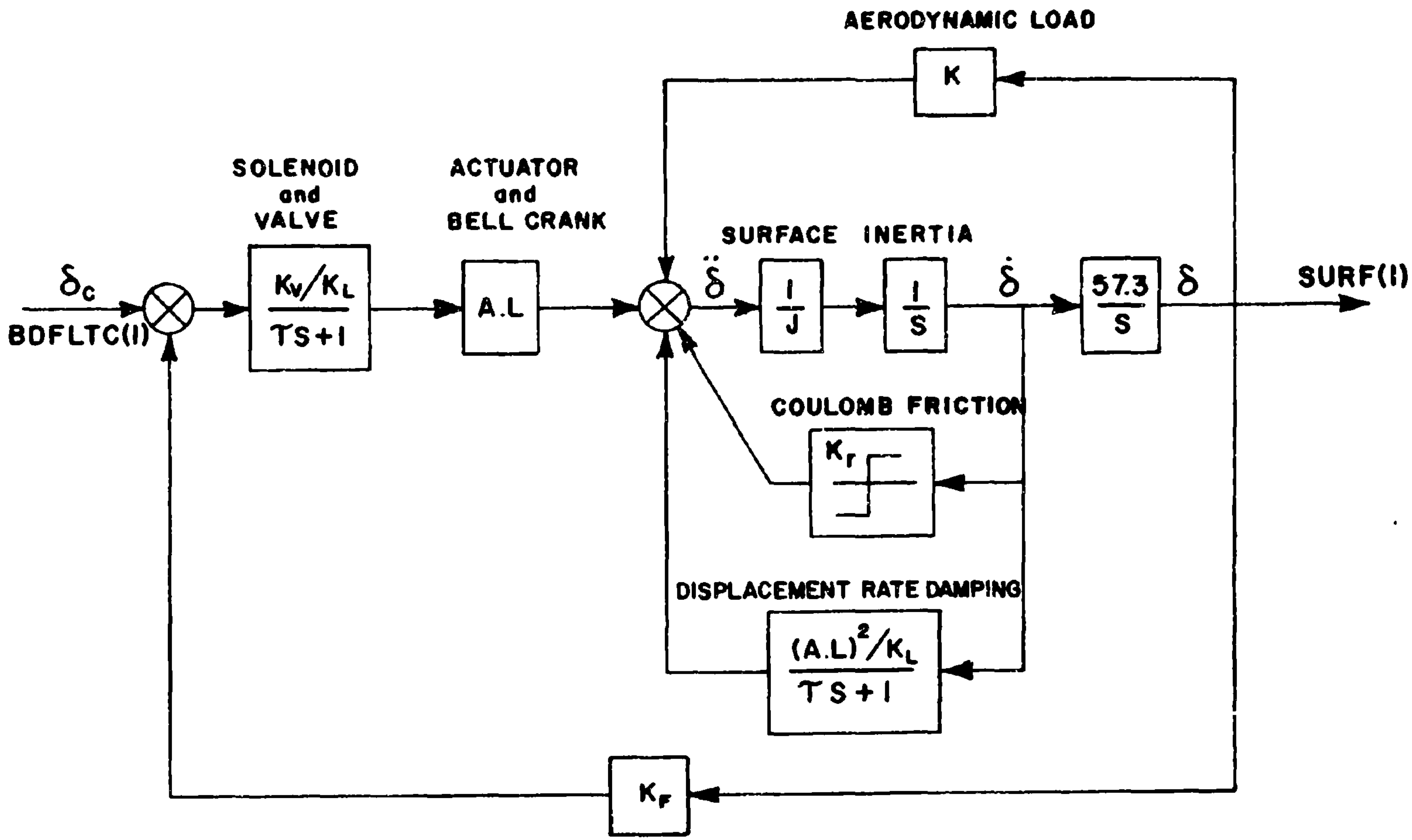


Figure 16. Actuator Position Loop Block Diagram

The transfer function for either the position feedback system or the torque balance system can be brought into the following form:

$$\frac{\delta}{\delta_c} = \frac{K_V A^* \ell (57.3)}{K_L [S(K_L JS(\tau S+1) + K_R(\tau S+1) + (A\ \ell)^2/K_L) + K_S K_\delta 57.3(\tau S+1)]}$$

and similarly for the position loop block diagram.

If either the torque balance system or the position loop control system is to be activated, then CKACT should be either set equal to one or zero, depending on whether the aerodynamic tables for FMH1, FMH2, FMH3, and FMH4 are included in the data tables. The variable BDMAX limits the maximum amplitude of the fin motion. The low frequency actuator equations are developed below.

Low Frequency Actuator

BDELT(1) = BDELT(1) $- \delta p + \delta q - \delta r$

BDELT(2) = BDELT(2) $- \delta p + \delta q + \delta r$

BDELT(3) = BDELT(3) $+ \delta p + \delta q - \delta r$

BDELT(4) = BDELT(4) $+ \delta p + \delta q + \delta r$

δ_1 = BDELT(1)

δ_2 = BDELT(2)

δ_3 = BDELT(3)

δ_4 = BDELT(4)

where

δ_p = DELTPB

δ_q = DELTQB

δ_r = DELTRB

The mechanization of these equations may be found in the Program Listing (Section V) for the low frequency actuator. The high frequency actuator program listing may be found in Appendix II.

2.9.1 Fin Deflection

A positive pitch rate (motion up) is obtained with a negative δq, where

$$\delta q = \frac{\delta_1 + \delta_2 + \delta_3 + \delta_4}{4}.$$

A positive roll rate (motion clockwise about the X body axis) is obtained with a positive δp, where

$$\delta p = \frac{(\delta_3 - \delta_2 + \delta_4 - \delta_1)}{4}.$$

A positive yaw rate (motion clockwise about the Z body axis) is obtained with a negative δr.

A positive surface deflection is defined as a trailing edge down. The surfaces are labeled by looking at the missile tail-on, with δ_1 being the upper right surface, δ_2 the lower right surface, δ_3 the lower left surface, and δ_4 the upper left surface, as shown in Figure 17.

It is assumed that the surface effectiveness will be given in terms of δq, δp, and δr as a function of α' and ϕ'. These terms will be considered as a part of the aerodynamic coefficients given in the primed axis system.

2.10 A3 - Engine Module

As a result of various sources of error occurring in the manufacture and assembly of a solid propellant motor, the thrust alignment is not perfect. The coordinate system used in determining the misalignment the user wishes to simulate is shown in Figure 18.

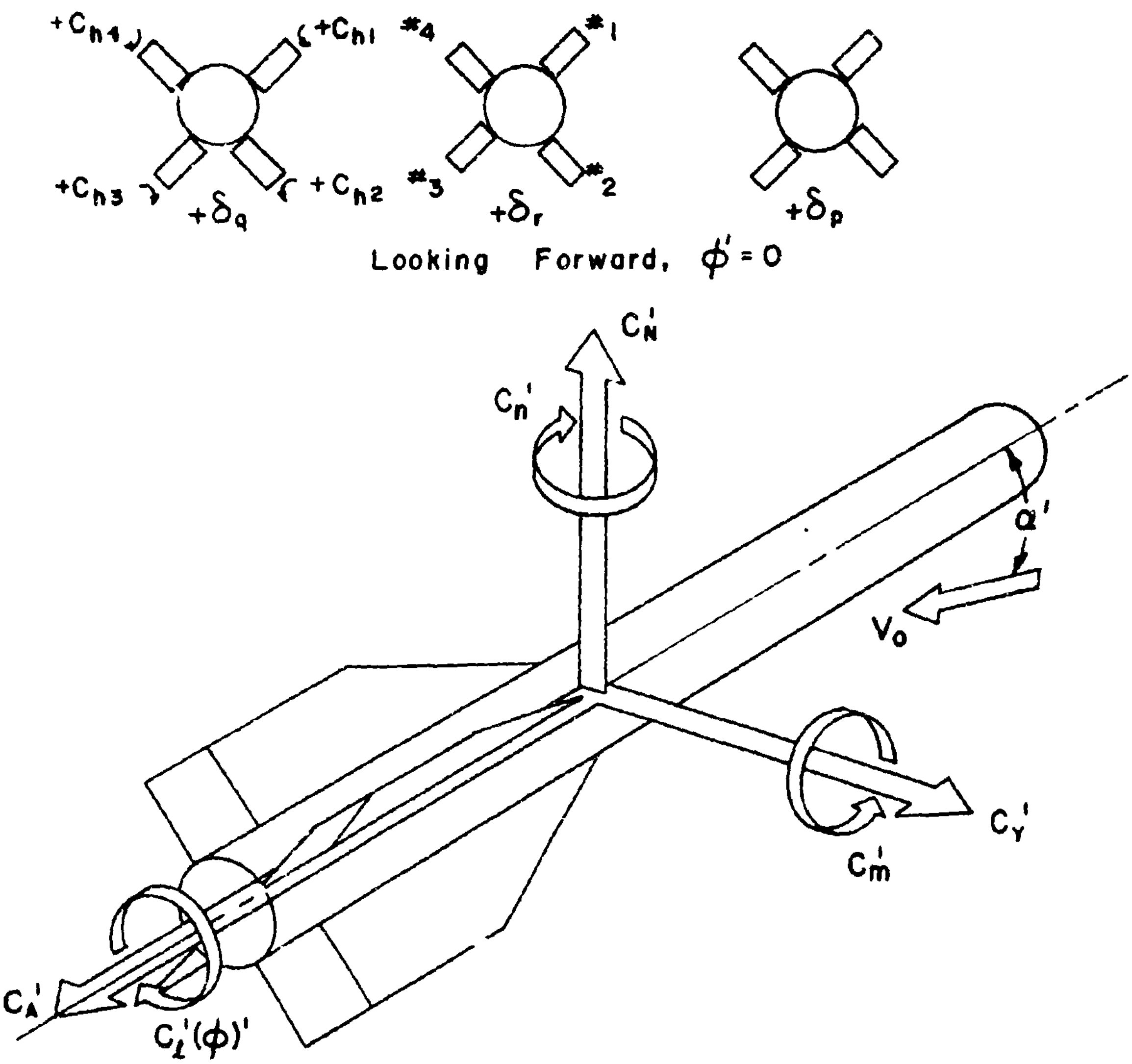
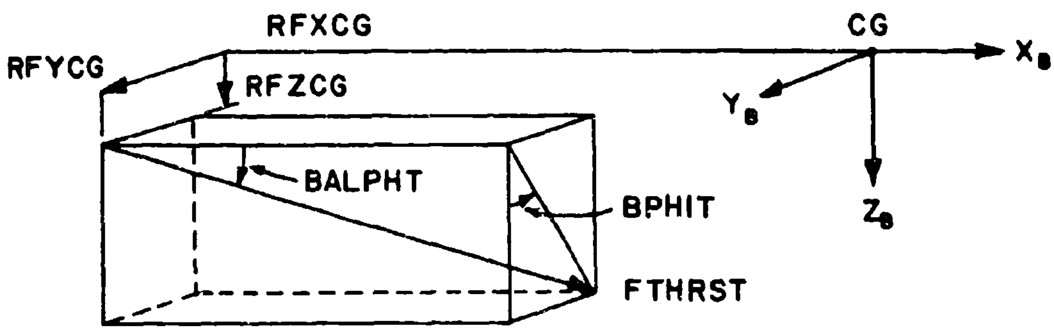


Figure 17. Fin Sign Conventions



RFXCG - X Component of thrust vector with respect to body axis in the X direction

RFYCG - Y Component of thrust vector with respect to body axis in the Y direction

RFZCG - Z Component of thrust vector with respect to body axis in the Z direction

FTHRST - Missile Thrust

Figure 18. Offset Thrust Vector Coordinate System

SECTION III

VARIABLE LOCATIONS

3.1 Variable Names, Block Locations, and Definitions

Since the proper use of this program requires that the definition of upwards of five hundred singly dimensioned variables as well as many multidimensioned be made, it is clear that some order must be maintained in the allocation of storage or serious programming difficulties could arise. Therefore, blocks of common location have been allocated to specific subroutines as shown in Table III. This procedure should be continued in the event it is necessary to add variables as a result of program modifications.

Of the large number of variables actually listed by the program, only two hundred and fifty appear to be significant in the preparation of the input or of an aid in understanding the output. Therefore, it was felt that they should be separately defined. This is done in Table IV. It should be noted that the units of the variables listed in that table should be considered to be in feet, seconds, pounds, or degrees unless otherwise specified.

TABLE III. BLANK COMMON ASSIGNMENTS

Array Index C( )	Module Name	Module Description
0 - 50	C10	Unsteady Illuminator
50 - 102	G2	Steady Winds, Variable Winds
200 - 299	G3	Air Data
350 - 399	G5	Coordinate Conversion
400 - 499	S1, S1I	Seeker - Platform
800 - 899	C1, C1I	Autopilot
1100 - 1149	C4, C4I	Actuators
1200 - 1299	A1	Aero Table Look-Up
1300 - 1399	A2	Forces and Moments
1400 - 1499	A3, A3I	Engine
1600 - 1699	D1, D1I	Translational Dynamics
1700 - 1799	D2, D2I	Rotational Dynamics

Note: Locations 1950 - 4310 are reserved for Executive Subroutines, Initial Conditions, and Input-Output Instructions.

TABLE IV. COMMON LOCATION, VARIABLE NAME, AND DEFINITION

Common Location	Variable Name	Definition
C(1)	BORE	Maximum boresight error
C(2)	WAND	Maximum pointing error
C(3)	RADIUS	Maximum deviation of hotspot from beam centroid
C(4)	HILL	Height of illuminator
C(5)	RILL	Ground range of illuminator
C(6)	AISPOT	0. - Centroid tracker 1. - Hotspot tracker
C(7)	AILL	0. - Stationary illuminator* 1. - Moving illuminator
C(8)	SPOTMO	0. - No spot motion 1. - Spot motion
C(9)	XSPOT	X - Coordinate of centroid or hotspot
C(10)	YSPOT	Y - Coordinate of centroid or hotspot
C(11)	AIFAC	0. - Tracker on ground or on launch aircraft 1. - Tracker on separate aircraft
C(12)	VILM	Velocity of illuminator, Mach number
C(18)	XILL	Ground range in X direction of illuminator

*Must give HILL, XILL for input

TABLE IV (Continued)

Common Location	Variable Name	Definition
C(51)	BPSIW	Mean angle of wind velocity vector
C(52)	VWTE	Mean wind velocity
C(53)	RHW	Altitude above which all the winds are zero
C(55)	SW	Standard deviation from mean angle of wind velocity vector BPSIW
C(56)	RWINC	Shear layer of wind. Depth of wind layers at which wind velocity and angle remain constant.
C(58)	SW1	Standard deviation from mean wind velocity VWTE
C(100)	VWXE	Wind velocity (X component with reference to the earth-fixed coordinate system)
C(101)	VWYE	Wind velocity (Y component with reference to the earth-fixed coordinate system)
C(102)	VWZE	Wind velocity (Z component with reference to the earth-fixed coordinate system)
C(203)	PDYNMC	Dynamic pressure
C(204)	VMACH	Mach number
C(205)	DRHO	Air density
C(206)	VSOUND	Speed of sound
C(207)	VAIRSP	Missile velocity with respect to air mass in earth axes

TABLE IV (Continued)

Common Location	Variable Name	Definition
C(208)	RHZRO	Initial altitude of the missile
C(209)	RH	Present altitude of the missile
C(350)	BTHT	Pitch angle in degrees - θ
C(351)	BPSI	Roll angle in degrees - ψ
C(352)	BPHI	Yaw angle in degrees - ϕ
C(356)	VTOTE	Total missile velocity
C(357)	BGAMH	Horizontal proportional navigation angle (degrees)
C(358)	BGAMV	Vertical proportional navigation angle (degrees)
C(367)	BALPHA	Vertical component of angle of attack
C(368)	BALPHY	Horizontal component of angle of attack
C(369)	BALPHP	($\alpha' = \sqrt{BALPHA^2 + BALPHY^2}$) total angle of attack
C(370)	BPHIP	ϕ' orientation of wind vector in roll axis
C(371)	RANGE	Range
C(372)	RXBA	Range (X component in body coordinate system)
C(373)	RYBA	Range (Y component in body coordinate system)
C(374)	RZBA	Range (Z component in body coordinate system)

TABLE IV (Continued)

Common Location	Variable Name	Definition
C(377)	BALPD	d(BALPHA)/dt
C(378)	BALYD	d(BALPHY)/dt
C(379)	BALPPD	d(BALPHP)/dt
C(380)	RANGO	The distance of missile from the launch point
C(403)	EZ	Seeker output to autopilot (pitch)
C(407)	EY	Seeker output to autopilot (yaw)
C(427)	BTHTG	Platform position (θ_g)
C(431)	BPSIG	Platform position yaw gimbal angle (ψ_g)
C(432)	RXG	Range X in gimbal axes
C(433)	RYG	Range Y in gimbal axes
C(434)	RZG	Range Z in gimbal axes
C(435)	BEPSZ	Angular position of the line of sight in gimbal axes (see Figure 8)
C(435)	BEPSY	
C(437)	WZ	Missile body rate W_Z
C(438)	WY	Missile body rate W_Y
C(441)	SZGBIS	Pitch gimbal torque bias (deg/sec)
C(442)	SYGBIS	Yaw gimbal torque bias (deg/sec)
C(443)	OPTKR	Optics routine
C(444)	OPTBKL	Optical breaklock
C(445)	UT	Time at which next pulse expected
C(446)	CDT	Pulse rate, sampling period

TABLE IV (Continued)

Common Location	Variable Name	Definition
C(450)	CSW	Acquisition gain seeker constant
C(451)	CAGE	[(>0) uncaged gimbals] [(<0) remain in caged position]
C(452)	QBREAK	(Breaklock has occurred due to loss of signal) (automatically parameterized)
C(453)	RBKLOK	Range at breaklock (maximum range at which lock-on can take place)
C(454)	BEDGE	Half the field of view
C(455)	WEPSMX	Breaklock drift rate
C(456)	CKSKR	Seeker gain
C(457)	CROSTP	Pitch to yaw friction coupling
C(458)	CROSPT	Yaw to pitch friction coupling
C(460)	GUIDE	(=1 missile guidance system in effect) (=0 missile guidance system not in effect)
C(461)	SAMP	Preprogrammed guidance trajectory (cutoff check automatically parameterized) (0 - missile uses preprogrammed flight path) (1 - missile uses preprogrammed flight path until seeker acquires target)
C(464)	CGAMVS	Vertical trajectory programming constant
C(465)	CGAMHS	Horizontal trajectory programming constant
C(466)	ZLAZR	Location of laser spot on target in X direction due to ground or airborne FAC

TABLE IV (Continued)

Common Location	Variable Name	Definition
C(467)	YLAZR	Location of laser spot on target in Y direction due to ground or airborne FAC
C(468)	DEFOCS	Half angle in degrees of angle intercepted by image of laser spot on detector surface
C(469)	DETRAD	Half angle in degrees of angle intercepted by quadrant detector
C(472)	CKSK1	Seeker driver constant
C(473)	VLAZRP	Used in pulse loss calculation
C(850)	HLIMO	Limit on δ_C from pitch and yaw plane (deg) (fins 1 and 3)
C(851)	HLIME	Limit on δ_C from pitch and yaw plane (deg) (fins 2 and 4)
C(852)	QBIAS	Pitch body rate bias (deg/sec) (used as "g" bias)
C(853)	RBIAS	Yaw body rate bias (deg/sec)
C(855)	GZ	Navigation ratio for pitch plane
C(856)	GY	Navigation ratio for yaw plane
C(863)	TAUZ	Pitch guidance lag filter (rad/sec)
C(864)	TAUY	Yaw guidance lag filter (rad/sec)
C(865)	TDY1	Rate loop gain switch 1 (sec)
C(866)	TDY2	Rate loop gain switch 2 (sec)
C(877)	TAUL	Guidance lead filter (rad/sec)
C(888)	CKSK2	Seeker gain constant

TABLE IV (Continued)

Common Location	Variable Name	Definition
C(1117)	BSURF 1	δ_1
C(1118)	BSURF 2	δ_2 fin deflection output for fins 1, 2, 3, 4
C(1119)	BSURF 3	δ_3
C(1120)	BSURF 4	δ_4
C(1160)	DELTPB	$\delta_p = (-\delta_1 - \delta_2 + \delta_3 + \delta_4)/4$
C(1161)	DELTQB	$\delta_q = (\delta_1 + \delta_2 + \delta_3 + \delta_4)/4$
C(1162)	DELTRB	$\delta_r = (-\delta_1 - \delta_2 + \delta_3 + \delta_4)/4$
C(1260)	CXERR	Drag coefficient error
C(1261)	CZERR	Normal force (C'_z) coefficient error
C(1262)	CYERR	Side force (C'_y) coefficient error
C(1263)	CLERR	Roll moment (C'_L) coefficient error
C(1264)	CMERR	Pitch moment (C'_M) coefficient error
C(1265)	CNERR	Yaw moment (C'_N) coefficient error
C(1300)	FXBA	The X component of aero force in body coordinate system
C(1301)	FYBA	The Y component of aero force in body coordinate system
C(1302)	FZBA	The Z component of aero force in body coordinate system
C(1303)	FMXBA	The X component of aero moment in body coordinate system
C(1304)	FMYBA	The Y component of aero moment in body coordinate system

TABLE IV (Continued)

Common Location	Variable Name	Definition
C(1305)	FMZBA	The Z component of aero moment in body coordinate system
C(1306)	RFAREA	Missile reference area (ft^2)
C(1307)	RFLGTH	Missile reference length (ft)
C(1308)	RDELCG	Center of gravity shift (ft)
C(1309)	FMH1	Hinge moments
C(1310)	FMH2	
C(1311)	FMH3	
C(1312)	FMH4	
C(1313)	RFXCG	Thrust vector displacements (ft)
C(1314)	RFYCG	
C(1315)	RFZCG	
C(1316)	RLUG	Distance between lugs (ft)
C(1317)	RAIL	Rail length (ft) (between rear of front lug and end of rail)
C(1320)	FMXTH	X component of moment caused by thrust misalignments
C(1321)	FMYTH	Y component of moment caused by thrust misalignments
C(1322)	FMZTH	Z component of moment caused by thrust misalignments
C(1323)	FMXLUG	X component of moment due to lugs
C(1324)	FMYLUG	Y component of moment due to lugs
C(1325)	FMZLUG	Z component of moment due to lugs

TABLE IV (Continued)

Common Location	Variable Name	Definition
C(1401)	BALPHT	The angles as indicated in Figure 18
C(1402)	LPHIT	
C(1403)	QNALGN	(> 0; include thrust misalignment angles)
C(1404)	FCFTH	Fractional increase in total thrust
C(1405)	QBURN	Parameterized by program
C(1410)	FTHRST	Missile thrust
C(1411)	FTHX	X component of missile thrust
C(1412)	FTHY	Y component of missile thrust
C(1413)	FTHZ	Z component of missile thrust
C(1414)	CISP	Specific impulse (lb/sec)
C(1415)	DWT	Total missile plus propellent wt (lb) initial
C(1416)	DWP	Propellent weight (lb)
C(1417)	RDCGO	Initial value of c.g. shift (ft)
C(1418)	RDCGF	Burnout value of c.g. shift (ft)
C(1419)	FMIXO	Initial value of moment of inertia about the roll axis (slugs ft)
C(1420)	FMIYO	Initial value of moment of inertia about the pitch axis (slugs ft)
C(1421)	FLCGA	Distance between launch c.g. and rear lug (ft)
C(1422)	RLCG	Present position of c.g. of missile

TABLE IV (Continued)

Common Location	Variable Name	Definition
C(1496)	UIMPD	Thrust of the motor
C(1499)	UIMP	Impulse of the motor
C(1603)	VXE	X,Y,Z coordinates of missile velocity with respect to the earth fixed coordinate system
C(1607)	VYE	
C(1611)	VZE	
C(1615)	RXE	X,Y,Z coordinates of missile c.g. with respect to the earth fixed coordinate system
C(1619)	RYE	
C(1623)	RZE	
C(1624)	AXBA	X component of acceleration in body coordinate axis
C(1625)	AYBA	Y component of acceleration in body coordinate axis
C(1626)	AZBA	Z component of acceleration in body coordinate axis
C(1627)	AGRAV	Gravitational constant
C(1628)	DMASS	Current mass
C(1629)	ATHRST	Target thrust
C(1630)	ATURNT	Maximum transverse acceleration of target in terms of g
C(1632)	VDELX	Relative velocity of missile to target in X direction
C(1633)	VDELY	Relative velocity of missile to target in Y direction
C(1634)	VDELZ	Relative velocity of missile to target in Z direction

TABLE IV (Continued)

Common Location	Variable Name	Definition
C(1635)	RDELX	Range difference between target and missile in the X direction of the earth fixed coordinate system
C(1636)	RDELY	Range difference between target and missile in the Y direction of the earth fixed coordinate system
C(1637)	RDELZ	Range difference between target and missile in the Z direction of the earth fixed coordinate system
C(1644)	ATARG	Acceleration of the target
C(1647)	VTARG	Velocity of the target
C(1648)	RTXED	The X component of the velocity of the target in the earth fixed coordinate system
C(1651)	RTXE	The X coordinate of the position of the target in the earth fixed coordinate system
C(1652)	RTYED	The Y component of the velocity of the target in the earth fixed coordinate system
C(1655)	RTYE	The Y coordinate of the position of the target in the earth fixed coordinate system
C(1656)	RTZED	The Z component of the velocity of the target in the earth fixed coordinate system
C(1659)	RTZE	The Z coordinate of the position of the target in the earth fixed coordinate system
C(1660)	VTXE	The X component of the velocity of the target in the earth fixed coordinate system

TABLE IV (Continued)

Common Location	Variable Name	Definition
C(1661)	VTYE	The Y component of the velocity of the target in the earth fixed coordinate system
C(1662)	VTZE	The Z component of the velocity of the target in the earth fixed coordinate system
C(1663)	VDXB	The X component of the acceleration of missile in the body coordinate axes
C(1664)	VDYB	The Y component of the acceleration of missile in the body coordinate axes
C(1665)	VDZB	The Z component of the acceleration of missile in the body coordinate axes
C(1666)	BDIVE	Initial pitch orientation of the aircraft (missile assumed oriented parallel to aircraft)
C(1667)	RSLANT	Initial slant range
C(1668)	RXO	The X component of the original launch point of the missile in the earth fixed coordinate system
C(1669)	RYO	The Y component of the original launch point of the missile in the earth fixed coordinate system
C(1670)	RZO	The Z component of the original launch point of the missile in the earth fixed coordinate system
C(1672)	BPSITD	The angular rate of turn of target
C(1675)	BPSIT	The total angle through which the target has turned in degrees

TABLE IV (Continued)

Common Location	Variable Name	Definition
C(1676)	ANGX	The X component of the acceleration of the missile in terms of g with respect to body axes
C(1677)	ANGY	The Y component of the acceleration of the missile in terms of g with respect to body axes
C(1678)	ANGZ	The Z component of the acceleration of the missile in terms of g with respect to body axes
C(1700)	CFA11D	Derivative of CFA11
C(1703)	CFA11	Cosψ Cosθ
C(1704)	CFA12D	Derivative of CFA12
C(1707)	CFA12	Sinψ Cosθ
C(1708)	CFA13D	Derivative of CFA13
C(1711)	CFA13	-Sinθ
C(1712)	CFA21D	Derivative of CFA21
C(1715)	CFA21	Sinψ Cosθ + Cosψ Sinθ Sinϕ
C(1716)	CFA22D	Derivative of CFA22
C(1719)	CFA22	Cosψ Cosϕ + Sinψ Sinθ Sinϕ
C(1720)	CFA23D	Derivative of CFA 23
C(1723)	CFA23	Cosθ Sinϕ
C(1724)	CFA31D	Derivative of CFA31
C(1727)	CFA31	Cosψ Sinθ Cosϕ + Sinψ Sinϕ
C(1728)	CFA32D	Derivative of CFA32

TABLE IV (Concluded)

Common Location	Variable Name	Definition
C(1731)	CFA32	Sinψ Sinθ Cosφ - Cosψ Sinφ
C(1732)	CFA33D	Derivative of CFA33
C(1735)	CFA33	Cosθ Cosφ
C(1736)	WPD	d(WP)/dt
C(1739)	WP	Roll rate of missile
C(1740)	WQD	d(WQ)/dt
C(1743)	WQ	Pitch rate of missile
C(1744)	WRD	d(WR)/dt
C(1747)	WR	Yaw rate of missile
C(1748)	FMIX	Missile moments of inertia about the X,Y, and Z missile body axes in flug-feet^2
C(1749)	FMIY	
C(1750)	FMIZ	
C(1751)	CRAD	Conversion factor (from radians to degrees)
C(1752)	BPHIO	Initial roll angle of missile
C(1753)	BTHTO	Initial pitch angle of missile
C(1754)	BPSIO	Initial yaw angle of missile

3.2 Subroutine Call Sequence

The subroutine call sequence is determined by the order in which these subroutines are identified in the data card assembly. A data card is identified by the program as a subroutine call by the number 2 located in column 2. The identification number of the subroutine may be called as either [MODNO(NOMOD)] or [XMODNO(NOMOD)] by the program. This integer must be right adjusted to column 25 on the card. Table V shows the identification number and the subroutines called in the example problem. If other routines are required, they will be found listed with their identification numbers in subroutine AUXSUB.

3.3 State Variables

The state variables within this six-degree-of-freedom simulation program are defined in the initialization subroutines (modules). These variables are identified through the IPL table which also defines the location of the state variables. Only these variables are integrated by the integration routine AMRK. Other variables found in the program which are derivatives are not state variables by this definition. A listing of the sequence number, IPL numbers, and variable names are found in Table VI. The listing is for the program when it contains the high frequency autopilot and actuator.

In the event a location is defined as a state variable, the following convention must be observed:

C(J + 3) State variable

then

C(J) is the derivative of that state variable.

TABLE V. INITIALIZATION SUBROUTINE CALL SEQUENCE
[(By Subroutine AUXI)(As defined by current program listing)]

NOMOD	MODNO(NOMOD) XMODNO(NOMOD)	SUBROUTINE CALL
1.	23	G2I
2.	24	G3I
3.	26	G5I
4.	28	S1I
5.	7	C1I
6.	10	C4I
7.	2	A1I
8.	4	A3I
9.	3	A2I
10.	17	D1I
11.	18	D2I

TABLE VI. STATE VARIABLES AND DERIVATIVES NAMES AND LOCATION CODES

Sequence No.	N	IPL(N)	C(IPL(N))	Module Location
1.	1	424	BTHTGD	S1, S1I
2.		427	BTHTG	SURFACE AND PLATFORM
3.	2	428	BPSIGD	
4.		431	BPSIG	
5.	3	800	BPHISD	C1, C1I
6.		803	BPHIS	HIGH FREQUENCY AUTOPILOT INITIALIZATION MODULE
7.	4	804	WQSDD	
8.		807	WQSP	
9.	5	808	WQSD	
10.		811	WQS	
11.	6	812	WRSDD	
12.		815	WRSP	
13.	7	816	WRSD	
14.		819	WRS	
15.	8	820	ESUMOD	
16.		823	ESUMO	
17.	9	824	ESUMED	
18.		827	ESUME	
19.	10	828	EZSDD	
20.		831	EZSP	
21.	11	832	EZSD	
22.		835	EZS	

TABLE VI (Continued)

Sequence No.	N	IPL(N)	C(IPL(N))	Module Location
23.	12	836	EYSDD	
24.		839	EYSP	Ci, C1I
25.	13	840	EYSD	HIGH FREQUENCY AUTOPILOT INITIALIZATION MODULE
26.		843	EYS	
27.	14	880	EZSSD	
28.		883	EZSS	
29.	15	884	EYSSD	
30.		887	EYSS	
31.	16	1100	BDELTD(1)	
32.		1103	BDELT(1)	C4, C4I
33.	17	1104	BDELTD(2)	HIGH FREQUENCY MODULE (ACTUATORS)
34.		1107	BDELT(2)	
35.	18	1108	BDELTD(3)	
36.		1111	BDELT(3)	
37.	19	1112	BDELTD(4)	
38.		1115	BDELT(4)	
39.	20	1124	BDLTDD(1)	
40.		1127	BDELTP(1)	
41.	21	1128	BDLTDD(2)	
42.		1131	BDELTP(2)	
43.	22	1132	BDLTDD(3)	
44.		1135	BDELTP(3)	
45.	23	1136	BDLTDD(4)	
46.		1139	BDELTP(4)	

TABLE VI (Continued)

Sequence No.	N	IFL(N)	C(IFL(N))	Module Location
47.	24	1496	LIMPD	A3I
48.		1499	LIMP	ENGINE
49.	25	1600	VXED	D1, D1I
50.		1603	VXE	TRANSLATIONAL DYNAMICS
51.	26	1604	VYED	
52.		1607	VYE	
53.	27	1612	RXED	
54.		1615	RXE	
55.	28	1616	RYED	
56.		1619	RYE	
57.	29	1620	RZED	
58.		1623	RZE	
59.	30	1640	VTARGD	
60.		1643	VTARG	
61.	31	1644	BPSITD	
62.		1647	BPSIT	
63.	32	1648	RTXED	
64.		1651	RTXE	
65.	33	1652	RTYED	
66.		1655	RTYE	
67.	34	1656	RTZED	
68.		1659	RTZE	

TABLE VI (Concluded)

Sequence No.	N	IPL(N)	C(IPL(N))	Module Location
69.	35	1700	CFA11D	D2, D2I
70.		1703	CFA11	ROTATIONAL DYNAMICS
71.	36	1704	CFA12D	
72.		1707	CFA12	
73.	37	1708	CFA13D	
74.		1711	CFA13	
75.	38	1712	CFA14D	
76.		1715	CFA14	
77.	39	1716	CFA22D	
78.		1719	CFA22	
79.	40	1720	CFA23D	
80.		1723	CFA23	
81.	41	1724	CFA310	
82.		1727	CFA31	
83.	42	1728	CFA32D	
84.		1731	CFA32	
85.	43	1732	CFA33D	
86.		1735	CFA33	
87.	44	1736	WPD	
88.		1739	WP	
89.	45	1740	WQD	
90.		1743	WQ	
91.	46	1744	WRD	
92.		1757	WR	

SECTION IV

INPUT REQUIREMENTS

4.1 Initial Conditions

In order to simplify the input data, the options which had existed in the original program have been eliminated. It is assumed by the program that all variables not initialized are automatically set equal to zero. Input data and initial conditions are entered into the program by entering a number 3 in column 2 of the data card which identifies the type of information. The name of the variable may be entered in column 3 to 20. Common location of the variable must be entered right adjusted in columns 21 to 25 and the numerical data in columns 31 to 45. Figure 19 shows the position of the data card in the completed program deck which is ready for submission, as well as the actual data card format.

In addition, since the seeker is generally assumed to lock on before launch, the gimbal angles are automatically initialized to this position. However, in the cases where gimbal angles must be chosen in any other position, the transformations and angular displacements between gimbal axes and body axes coordinate systems are given in Appendix I and Figure I-1, respectively.

Initial position and velocity can only be specified in one manner for simplicity. They are specified in terms of the following variables:

BDIVE (in degrees, negative when orientation below horizontal)

RSLANT ([illegible]) (feet)

BALPHA (367) (degrees)

BALPHY (368) (degrees)

VMACH (204) (Mach number)

It should be noted that the program when used to simulate many missions requires only that subsequent changes in data be added since the program will only update the last data set for the next run. (See "Program Description.")

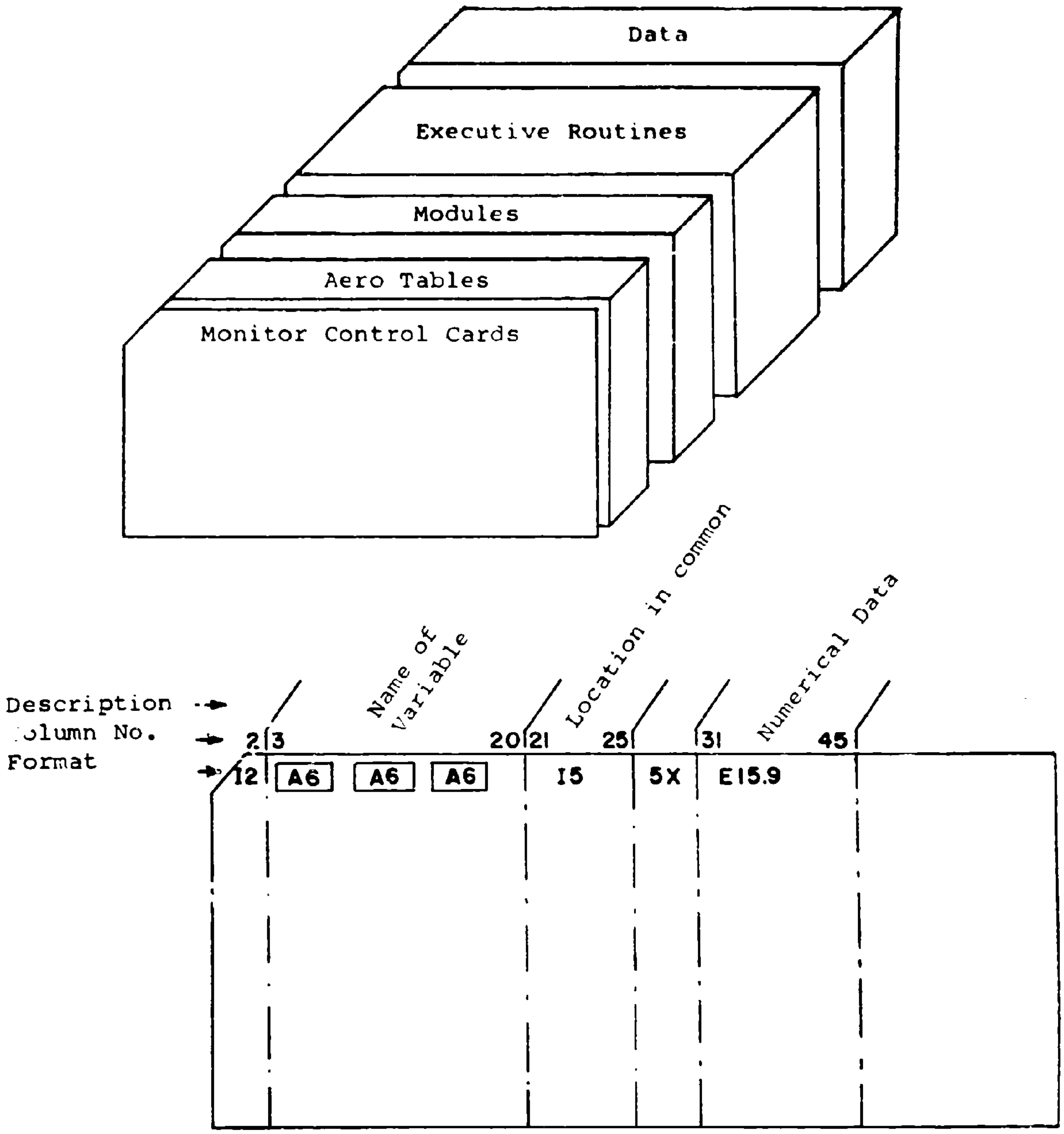


Figure 19. Data Card Formats and Deck Setup

SECTION V

PROGRAM LISTING

5.1 Complete Six-Degree-of-Freedom Program Listing with Example

```
C
C*****DIPOCS TO BE USED WITH FORTRAN AMRK INTEGRATION ROUTINE
C
      CCMMON C(4310),GRAPH
      EQUIVALENCE (C(2662),HMIN  ), (C(2663),HMAX  ), (C(2664),DER   ),
C                 (C(2561),N     ), (C(2562),IPL   ), (C(2965),VAR   ),
C                 (C(2000),T     ), (C(2011),KSTEP ), (C(2010),STEP  ),
C                 (C(2012),LSTEP ), (C(2008),PLOTNO), (C(2009),NCPLOT),
C                 (C(2023),OPOINT), (C(2025),TIME  ), (C(2325),VLABLE),
C                 (C(3167),NOOUT ), (C(2022),OPTN10), (C(2006),REPPLT),
C                 (C(2865),EU    ), (C(2765),EL    ), (C(2007),PTLESS)
      EQUIVALENCE                   (C(1971),RITE  ), (C(1972),RKUTTA)
      EQUIVALENCE (C(1973),KASE  ), (C(1974),NJ    ), (C(1975),NPT   )
      DIMENSION GRAPH(2,2),TIME(2)
      DIMENSION   VLABLE(2,15)     , IPL(100)        , DER(101)
      DIMENSION   VAR(101)         , EL(100)         , EU(100)
      EQUIVALENCE (C(1980),RN    )
      EQUIVALENCE (C(1981),RNT   )
      EQUIVALENCE (C(1982),PLOTN4)
      EQUIVALENCE (C(1983),PLOTN2)
      EQUIVALENCE (C(1984),NPLOT )
      INTEGER CPOINT
      INTEGER PLOTNO
C     INTEGER CPT
      EXTERNAL AUXSUB
      DO 22I=1,4310
   22 C(I)=0.
      NPT=2
      CALL CCUNTY
      CALL ZERO
 1000 CONTINUE
 1001 IF(PLOTNO.LE.0.)GOTO7
      IF(REPPLT.GT.0.)GOTO7
C
C                             REPPLT = 0.  USE NEW NO.4,7 (DISCARD OLD)
C                                      1.  USE OLD PLUS THOSE ADDED
C                                     -1.  USE NEW NO. 7 (DISCARD OLD)
      IF (REPPLT.GT.-1.0) NOOUT = U
      NPLOT=0
    7 CALL CINPT1
      KASE=0
      IF(RKUTTA.GT.0.0) NPT=1
      LSTEP = STEP
      NPLOT4=PLOTN4
      NPLOT2=PLOTN2
      NCPLOT=PLOTNO
 1002 CALL  SUBL1
 1003 CALL  AUX1
 1004 CALL  SUBL2
 1005 DO 60 I = 2,N
      J = [illegible]-1)
      EL(I-[illegible](J+1)
      EU(I-1)=C(J+2)
      VAR(I) = C(J+3)
   60 DER(I) = C(J)
      VAR(1) = T
 1006 CALL   AUXSUB
 1007 NJ=N-1
      CALL AMRK(AUXSUB)
 1008 DO 50 I = 2, N
      J = IPL(I-1)
   50 C(J+3) = VAR(I)
```

```
      T = VAR(1)
 1009 CALL  SUBL3
      IF ( KSTEP .EQ. 1 ) GO TO 1007
      DO 155 JV=2,N
  155 VAR(JV)=0.
      CALL S8
      CALL PROCES
      CALL RESET
      IF(LSTEP.EQ.5.OR.LSTEP.EQ.7.OR.NOPLOT.EQ.0)GOTO5
      CALL TIMEV(DELT)
      WRITE(6,96)DELT
   96 FORMAT(1H ,17HSTART PLOTTING ATF14.7)
      LESSPT=FTLESS
      OPOINT=OPOINT-LESSPT
      CALL PLOT4 (GRAPH,OPOINT,VLABLE,TIME,NPLOT4)
      CALL PLOT2 (NPLOT2)
      CALL PLOTN (NOPLOT)
      CALL TIMEV(DELT)
      WRITE(6,97)DELT
   97 FORMAT(1H ,18HPLOTTING ENDED AT F14.7)
      IF((RNT.GT.0.1 .AND. RN.EQ.RNT).AND. LSTEP.EQ.2)GO TO 70
    5 GO TO (1000,1001,1002,1003,1004,1005,1006,1007,1008,1009,1010),
     1 LSTEP
 1010 IF(CPTN10.GT.0.)
     1WRITE(6,6)(I,C(I),C(I+1),C(I+2),C(I+3),C(I+4),C(I+5),C(I+6),I=1,35
     *10,7)
    6 FORMAT(1H1/(I5,1P7E15.7))
   70 CALL S8
      CALL EXIT
      END
      BLOCK DATA
      COMMON /NCXCP/NCX(6)
     *        /CXARG/ALP(7),AM(8)
     *        /CXFUN/CX(56)
      DATA NCX/7,8,0,0,0,0/
      DATA ALP/0.,4.,6.,8.,10.,15.,20./
      DATA AM/0.,.7,.9,1.025,1.2,1.5,2.0,2.3/
      DATA CX/
     1 0.0,0.0,.18,.32,.66,1.56,2.66,
     2 0.0,0.0,.18,.32,.66,1.56,2.66,
     3 0.0,0.0,.22,.35,.71,1.7,3.27,
     4 0.0,0.0,.23,.37,.79,1.8,3.2,
     5 0.0,0.0,.21,.34,.78,1.65,3.26,
     6 0.0,0.0,.13,.25,.51,1.11,2.08,
     7 0.0,0.0,.10,.21,.38,.95,1.67,
     8 0.0,0.0,.09,.19,.35,.82,1.57/
      END
      BLOCK DATA
      COMMON /NCZ/NCN(4)
     *       /CZARG/ALP(8),AM(9)
     *       /CZFUN/CN(72)
      COMMON/NCCZ/NCC(4)
     *       /CCARG/ALPC(6),AMC(6)
     *       /CCZFUN/DCN(36)
      COMMON/NCM/NCM(4)
     *       /CMARG/ ALF(8),BM(9 )
     *       /CMFUN/CM(72)
      COMMON/NDCM/NDM(4)
     *             /CCMARG/ALFD(6),BMD(6)
     *             /CCMFUN/DCM(36)
      DATA  NCN/8,9,0,0/
      DATA ALP/0.,2.,4.,6.,8.,10.,15.,20./
```

```
      DATA AM/0.,.7,.9,1.05,1.1,1.5,1.7,2.0,2.3/
      DATA CN/
     * 0.0,.65,1.38,2.21,3.12,4.13,7.08,10.71,
     * 0.0,.66,1.41,2.25,3.17,4.19,7.18,10.84,
     * 0.0, .69, 1.46, 2.32, 3.27, 4.31, 7.36, 11.11,
     * 0.0, .77, 1.62, 2.56, 3.59, 4.72, 7.98, 12.02,
     * 0.0, .77, 1.62, 2.57, 3.60, 4.74, 8.02, 12.11,
     * 0.0,.71, 1.5, 2.39, 3.36, 4.43, 7.6 , 11.60,
     * 0.0, .7, 1.49, 2.37, 3.34, 4.40, 7.59, 11.61,
     * 0.0,.69, 1.47, 2.34, 3.30, 4.36, 7.56, 11.61,
     * 0.0,.67, 1.43, 2.27, 3.21, 4.26, 7.42, 11.47/
      DATA NCC/6,6,0,0/
      DATA ALPC/0., 4., 8., 12., 16., 20./
      DATA AMC/0.0,.7, .9, 1.1, 1.5, 2.3/
      DATA CCN/
     * 0.0, .06, .30, .75, 1.40, 2.20,
     * 0.0, .06, .30, .75, 1.40, 2.20,
     * 0.0, .06, .30, .79, 1.45, 2.26,
     * 0.0, .06, .30, .79, 1.45, 2.26,
     * 0.0, .06, .30, .77, 1.43, 2.24,
     * 0.0, .06, .30, .76, 1.41, 2.22/
      DATA NCM/8,9,0,0/
      DATA ALF/0.,2., 4., 6., 8., 10., 15., 20./
      DATA BM/0.0, .7, .9, 1.05, 1.1, 1.5, 1.., 2.0, 2.3/
      DATA CM/
     * 0.0, -1.16, -2.48, -3.97, -5.62, -7.51, -12.83, -19.28,
     * 0.0, -1.19, -2.54, -4.06, -5.76, -7.68, -13.10, -19.64,
     * 0.0, -1.25, -2.66, -4.23, -5.98, -7.97, -13.54, -20.26,
     * 0.0, -1.59, -3.35,-5.29, -7.40,-9.75, -16.24, -23.87,
     * 0.0, -1.66, -3.48, -5.48, -7.66, -10.09, -16.75, -24.56,
     * 0.0, -1.07, -2.31, -3.74, -5.35, -7.22, -12.48, -18.91,
     * 0.0, -1.11, -2.39, -3.85, -5.50, -7.40, -12.70, -19.12,
     * 0.0, -1.15, -2.47, -3.97, -5.65, -7.57, 12.90, -19.29,
     * 0.0, -1.14, -2.46, -3.96, -5.63, -7.54, -12.81, -19.09/
      DATA NCM/6,6,0,0/
      DATA ALFD/0., 4., 8., 12., 16., 20./
      DATA BMC/0., .7, .9, 1.1, 1.5, 2.3/
      DATA CCM/
     * 0.0, -.13, -.64, -1.82, -3.45, -5.40,
     * 0.0, -.13, -.64, -1.82, -3.45, -5.40,
     * 0.0, -.13, -.73, -1.93, -3.65, -5.65,
     * 0.0, -.14, -.77, -1.99, -3.70, -6.00,
     * 0.0, -.13, -.73, -1.93, -3.65, -5.65,
     * 0.0, -.13, -.73, -1.93, -3.65, -5.65/
      END
      BLOCK DATA
      COMMON /NCXO/NCXO(2)
     *       /CXOARG/AM(8)
     *       /CXOFUN/CXO(8)
      DATA NCXC/8,1/
      DATA AM /0.,.7,.9,1.025,1.2,1.5,2.0,2.3/
      DATA CXO/.24,.24,.23,.68,.76,.91,1.0,.94/
      END
      BLOCK DATA
      COMMON/ NCN2/NCN2(4)
     *       / CN2ARG/ALP(6), AM(3)
     *       / CN2FUN/CN2(18)
      DATA NCN2/6,3,0,0/
      DATA ALP/0., 4., 8., 12., 16., 20./
      DATA AM /0., .7, 2.3/
      DATA CN2/
     * 0.0, .17, .68, 1.50, 2.60, 3.90,
```

```
*  0.0, .17, .68, 1.50, 2.60, 3.90,
*  0.0, .17, .68, 1.50, 2.60, 3.90/
      END
      BLOCK DATA
      COMMON  /NCY2/NCY2(4)
*             /CY2ARG/ALP(6), AM(3)
*             /CY2FUN/CY2(18)
      DATA  NCY2/6,3,0,0/
      DATA  ALP /0., 4., 8., 12., 16., 20./
      DATA AM/0.0,.7,2.3/
      DATA  CY2 /
*  0.0, -.07, -.30, -.65, -1.1, -1.6,
*  0.0, -.07, -.30, -.65, -1.1, -1.6,
*  0.0, -.07, -.30, -.65, -1.1, -1.6/
      END
      BLOCK DATA
      COMMON/ NCL2/ NCL2(4)
*           /CL2ARG/ALP(6),AM(4)
*           /CL2FUN/CL2(24)
      DATA NCL2/6,4,0,0/
      DATA ALP/0.0, 4., 8., 12., 16., 20./
      DATA AM/0., .7, .9, 2.3/
      DATA CL2/
*  0.0, .02, .07, .14, .25, .33,
*  0.0, .02, .07, .14, .25, .33,
*  0.0,.03, .12, .275, .515,  .35,
*  0.0, .02, .07, .14, .25, .33/
      END
      BLOCK DATA
      COMMON/NCL3/NCL3(4)
*           /CL3ARG/ALP(6), AM(3)
*           /CL3FUN/CL3(18)
      DATA NCL3/6,3,0,0/
      DATA ALP/0., 4., 8., 12., 16., 20./
      DATA AM/0., .7, 2.3/
      DATA CL3/
*  0.0,.022,.043, .08, .143, .215,
*  0.0,.022,.043, .08, .143, .215,
*  0.0,.022,.043, .08, .143, .215/
      END
      BLOCK DATA
      COMMON/NCZD/NCN(8)
*           /CZCFUN/CN1(32),CN2(32),CN3(32),CN4(32),CN5(32), CN6(32),
*            CN7(32),CN8(32)
*           /CZCARG/ALP(8),DEF(4),AM(8)
      DATA NCN/8,4,8,0,0,0/
      DATA ALP/0., 2., 4., 6., 8., 10., 15., 20./
      DATA  DEF /-20., -10., 10., 20./
      DATA AM/0. .67, .9, 1.05, 1.2, 1.5, 2.,2.3/
      DATA CN1/
* -.222,-.218,-.215,-.212,-.208,-.204,-.194,-.181,
* -.205,-.204,-.203,-.201,-.198,-.196,-.190,-.181,
*  .206, .207, .208, .208, .208, .209, .208, .207,
*  .222, .224, .227, .229, .231, .233, .235, .237/
      DATA CN2/
* -.222,-.218,-.215,-.212,-.208,-.204,-.194,-.181,
* -.206,-.204,-.203,-.201,-.198,-.196,-.190,-.181,
*  .206, .207, .208, .208, .208, .209, .208, .207,
*  .222, .224, .227, .229, .231, .233, .235, .237/
      DATA CN3/
* -.252,-.249,-.245,-.241,-.237,-.232,-.220,-.207,
* -.236,-.235,-.233,-.231,-.228,-.225,-.217,-.209,
```

```
*  .236, .237, .239, .239, .240, .240, .240, .236,
*  .252, .255, .258, .260, .262, .265, .268, .269/
 DATA CN4/
* -.276,-.273,-.269,-.264,-.260,-.255,-.241,-.226,
* -.260,-.259,-.257,-.254,-.251,-.249,-.239,-.229,
*  .260, .261, .262, .263, .264, .264, .264, .261,
*  .276, .279, .282, .285, .288, .289, .294, .295/
 DATA CN5/
* -.246,-.243,-.240,-.238,-.232,-.227,-.215,-.202,
* -.230,-.229,-.227,-.225,-.222,-.220,-.212,-.203,
*  .230, .232, .232, .234, .234, .234, .234, .231,
*  .246, .249, .252, .255, .257, .258, .262, .263/
 DATA CN6/
* -.168,-.165,-.163,-.161,-.158,-.155,-.146,-.138,
* -.152,-.151,-.149,-.149,-.147,-.145,-.139,-.134,
*  .152, .152, .153, .153, .154, .154, .154, .153,
*  .168, .170, .172, .173, .175, .176, .179, .180/
 DATA CN7/
* -.131,-.129,-.127,-.125,-.123,-.121,-.115,-.108,
* -.115,-.114,-.113,-.112,-.111,-.109,-.106,-.102,
*  .115, .116, .116, .116, .116, .117, .116, .115,
*  .131, .132, .134, .135, .136, .137, .139, .140/
 DATA CN8/
* -.123,-.122,-.120,-.118,-.116,-.114,-.108,-.106,
* -.107,-.106,-.106,-.105,-.104,-.103,-.099,-.095,
*  .107,.108, .108, .109, .109, .109, .109, .107,
* .123, .125, .126, .127, .128, .129, .131, .132/
 END
 BLOCK DATA
 COMMON  /NCLC/NCL(6)
*          /CLCARG/ALF(6), DEF(4), AM(3)
*          /CLCFUN/CL1(24), CL2(24), CL3(24)
 DATA  NCL /6,4,3,0,0,0/
 DATA  ALP /0., 4., 8., 12., 16., 20./
 DATA  DEF /-20., -10., 10., 20./
 DATA AM/0.0,.17,2.3/
 DATA  CL1 /
*-.144,-.149,-.141,-.138,-.138,-.134,
*-.131,-.128,-.125,-.124,-.123,-.121,
* .131, .128, .125, .124, .123, .121,
* .144, .149, .141, .138, .138, .134/
 DATA  CL2 /
*-.144,-.149,-.141,-.138,-.138,-.134,
*-.131,-.128,-.125,-.124,-.123,-.121,
* .131, .128, .125, .124, .123, .121,
* .144, .149, .141, .138, .138, .134/
 DATA  CL3 /
*-.148,-.145,-.143,-.140,-.138,-.136,
*-.144,-.138,-.136,-.133,-.130,-.130,
* .144, .138, .136, .133, .130, .130,
* .148, .145, .143, .140, .138, .136/
 END
 BLOCK DATA
 COMMON /NCMC/NCM(6)
*         /CMCARG/ALP(8),DEF(4),AM(8)
*/CMCFUN/CM1(32),CM2(32),CM3(32),CM4(32),CM5(32),CM6(32),CM7(32),
*CM8(32)
 DATA NCM /8,4,8,0,0,0/
 DATA ALP/0.,2.,4.,6.,8.,10.,15.,20./
 DATA  DEF /-20., -10., 10., 20./
 DATA AM/0.,.7,.9,1.05,1.2,1.5,2.,2.3/
 DATA CM1/
```

```
1  .938, .937, .937, .938, .938, .939, .944, .948,
2  .864, .864, .864, .865, .864, .865, .870, .874,
3 -.864,-.864,-.864,-.865,-.864,-.865,-.870,-.874,
4 -.938,-.937,-.937,-.938,-.938,-.939,-.944,-.948/
 DATA CM2/
1 -.938,-.937,-.937,-.938,-.938,-.939,-.944,-.948,
2 -.864,-.864,-.864,-.865,-.864,-.865,-.870,-.874,
3  .864, .864, .864, .865, .864, .865, .870, .874,
4  .938, .937, .937, .938, .938, .939, .944, .948/
 DATA CM3/
1 -1.123,-1.137,-1.137,-1.138,-1.138,-1.14,-1.144,-1.152,
2 -1.064,-1.064,-1.064,-1.064,-1.064,-1.065,-1.072,-1.077,
3  1.064, 1.064, 1.064, 1.064, 1.064, 1.065, 1.072, 1.077,
4  1.123, 1.137, 1.137, 1.138, 1.138, 1.14, 1.144, 1.152/
 DATA CM4/
1 -1.277,-1.277,-1.277,-1.277,-1.277,-1.279,-1.285,-1.29,
2 -1.204,-1.203,-1.203,-1.203,-1.204,-1.205,-1.211,-1.216,
3  1.204, 1.203, 1.203, 1.203, 1.204, 1.205, 1.211, 1.216,
4  1.277, 1.277, 1.277, 1.277, 1.277, 1.279, 1.285, 1.29/
 DATA CM5/
1 -.796,-.793,-.791,-.789,-.788,-.787,-.787,-.786,
2 -.720,-.718,-.715,-.715,-.713,-.712,-.711,-.711,
3  .720, .718, .715, .715, .713, .712, .711, .711,
4  .796, .793, .791, .789, .788, .787, .787, .786/
 DATA CM6/
1 -.621,-.620,-.619,-.617,-.615,-.614,-.612,-.613,
2 -.545,-.544,-.543,-.541,-.540,-.539,-.537,-.534,
3  .545, .544, .543, .541, .540, .539, .537, .534,
4  .621, .620, .619, .617, .615, .614, .612, .613/
 DATA CM7/
1 -.586,-.584,-.583,-.582,-.580,-.579,-.577,-.578,
2 -.510,-.508,-.507,-.506,-.505,-.504,-.502,-.503,
3  .510, .508, .507, .506, .505, .504, .502, .503,
4  .586, .584, .583, .582, .580, .579, .577, .578/
 DATA CM8/
1 -.586,-.584,-.583,-.581,-.580,-.579,-.578,-.575,
2 -.510,-.508,-.507,-.506,-.505,-.504,-.502,-.503,
3  .510, .508, .507, .506, .505, .504, .502, .503,
4  .586, .584, .583, .581, .580, .579, .578, .575/
 END
 BLOCK DATA
 COMMON  /NTH/NTH(2)
*        /THARG/THA(20)
*        /THFUN/THF(20)
 DATA NTH/13,1/
 DATA THA/
* 0., .05, .1, .2, .4, .8, 1.3, 1.5, 1.6, 1.8, 2.0, 2.8, 100./
 DATA THF/
* 1., 5000., 4000., 3800., 3850., 3800., 3600., 2000.,1100.,
* 700., 250., 0.0, 0.0/
 END
 BLOCK DATA
 COMMON /NCLP/NL(4)
*       /CLPARG/ALP(8),AMP(7)
*      /CLPFUN/CLP(56)
 COMMON /NCMQ/NQ(4)
*       /CMQARG/ALQ(6),AMQ(4)
*      /CMQFUN/CMQ(24)
 DATA NL/8,7,0,0/
 DATA ALP/0.,2.,4.,6.,8.,10.,15.,20./
 DATA AMP/0.,.7,.9,1.05,1.5,2.0,2.3/
 DATA CLP/
```

```
1 -6.07,-6.40,-7.07,-7.73,-8.38,-9.50,-10.56,-11.31,
2 -6.07,-6.40,-7.07,-7.73,-8.38,-9.50,-10.56,-11.31,
3 -6.53,-6.86,-7.52,-8.19,-8.84,-9.95,-11.03,-11.79,
4 -7.31,-7.65,-8.32,-8.99,-9.65,-10.79,-12.0,-12.87,
5 -5.01,-5.35,-6.03,-6.72,-7.42,-8.75,-10.16,-11.18,
6 -4.54,-4.88,-5.55,-6.23,-6.96,-8.39,-9.84,-10.89,
7 -4.45,-4.79,-5.46,-6.15,-6.92,-8.37,-9.86,-10.94/
  CATA NC/6,4,0,0/
  CATA ALC/0.,4.,8.,12.,16.,20./
  CATA APC/0.,.17,.9,2.3/
  CATA CPC/
 1-.24,-.39,-.53,-.62,-.67,-.67,
 2-.24,-.39,-.53,-.62,-.67,-.67,
 3 -.25,-.41,-.55,-.66,-.72,-.75,
 4 -.25,-.41,-.55,-.66,-.72,-.75/
  ENC
  SUBRCUTINE CUADET(CETRIC,DEFICS,AA,BB,CC,DD,BEPSZ,BEPSY)
  CCMMCNC(4310)
  EGUIVALENCE(C(371),RT)
    CIMENSICNC(20),A(20),B(20)
C     DETRAC-HALF FOV OF CETECTOR
C     CEFRAC-HALFFCV  OF CEFOCUS SPOT
C     AA-LCWER LEFT QUACRNT
C     BB-LCWER RT CUACRANT
C     CC-UPPER LEFT QUACRNT
C     DC-UPPER RT CUACRNT
C     CEFCCS-HALF SIZE-OF DEFOCUS SPOT AT INFINITY
      LT=16
      CEFCCS=CEFICS/57.29578
      RT1=(1.+100./RT)
3     XOF=BEPSY/57.29578
      YCF=BEPSZ/57.29578
      VCF=.7071*(XCF+YOF)
      UCF=-.7071*(YOF-XOF)
      XCF=UCF
      YCF=VCF
      DETRAC=CETRIC/57.29578
      ZCF=SCRT(XOF**2+YOF**2)
      CELRAC=CETRAC+DEFOCS*RT1
      IF(ZCF.CE.CELRAD)GOTO42
      C2TR=CETRAD**2
      DEFRAC=GEFOCS*RT1
      D2FRAC=CEFRAC**2
      LN=LT/2
      DELF=CETRAC/LN
      CZCN=CELF/4.
      LC1=LT+1
      CC10I=1,LC1
      A(I)=0.
10    B(I)=0.
      CC30I=1,LC1
      CIM=CETRAC-(I-1)*DELF
      EXCET=SCRT(C2TR  -(CIM)**2)
      IF((D2FRAC-(CIM+YCF)**2).LT.0.)GOTO30
      XX=SCRT(C2FRAD-(CIM+YOF)**2)
      CO20NT=1,2
      X1=XCF-(-1.)**NT*XX
      IF(ABS(X1).GT.EXDET)D(NT)=SIGN(EXDET,X1)
20    IF(ABS(X1).LE.EXCET)D(NT)=(X1)
      IF(SIGN(1.,C(1)).EC.SIGN(1.,D(2)))GOTO22
      IF(C(1).LE.0.)GOTO35
      A(I)=C(1)-DZCN
```

```
      IF(A(I).LT.0.)A(I)=0.
      B(I)=C(2)+DZCN
      IF(B(I).GT.0.)B(I)=0.
      GCTC30
   35 A(I)=D(2)-DZCN
      IF(A(I).LT.0.)A(I)=0.
      B(I)=C(1)+CZCN
      IF(B(I).GT.0.)B(I)=0.
      GC TO 30
22    IF(C(1)+C(2))23,24,25
23    B(I)=-ABS(D(1)-C(2))
      A(I)=0.
      GCTC30
24    A(I)=0.
      B(I)=0.
      GCTC30
25    A(I)=ABS(C(1)-D(2))
      B(I)=0.
30    CCNTINUE
      AA=0.
      BB=0.
      CC=0.
      CC=0.
      A(LN+1)=0.
      B(LN+1)=0.
      DC40I=1,LN
      AA=ABS((A(I)+A(I+1))*.5*DELF)+AA
      BB=ABS((B(I)+B(I+1))*.5*DELF)+BB
      CC=ABS((A(I+LN)+A(I+LN+1))*.5*DELF)+CC
40    CD=ABS((B(I+LN)+B(I+LN+1))*.5*CELF)+DD
      ACET=3.14159*C2TR*.9
      TOTAL=AA+BB+CC+CC
      IF(TOTAL.GE.ADET)WRITE(6,51)RT
51    FCRMAT(15H  BLIND RANGE   ,1PE12.4)
      GCTC60
42    WRITE(6,50)RT
      CBREAK=-1.
      AA=0.
      BB=0.
      CC=0.
      CD=0.
50    FORMAT(32H  BREAKLCCK AT RANGE EQUAL TO    ,1PE12.4)
60    RETURN
      ENC
      SUBRCUTINEGAUSS(S,AM,V)
      CCMMCNC(4310)
      ECUIVALENCE(C(103),YFL)
      ECUIVALENCE(C(2000),T)
C     S-THE RECUIRED STANCARD DEVIATION
C     AM-IS THE RECUIRED MEAN
C     V-VALUE CF CCMPUTEC NORMALLY DISTRIBUTED RANDOM NUMBER
      A=0.
      B=1.
      CC 50 I=1,12
      YFL=RCNG(B)
   50 A=A+YFL
      V=(A-6)*S+AM
      RETURN
      ENC
      SUBRCUTINE C101
      COMMCN C(4310)
      ECUIVALENCE(C(7),AILL)
```

```
      EQUIVALENCE(C(11),AIFAC)
      EQUIVALENCE(C(12),VILM)
      EQUIVALENCE(C(204),VMACH)
      EQUIVALENCE(C(18),XIL)
      EQUIVALENCE(C(1615),RXE)
      EQUIVALENCE(C(4),HILL)
      EQUIVALENCE(C(1623),RZE)
      EQUIVALENCE(C(5),RILL)
      EQUIVALENCE(C(13),VILF)
      EQUIVALENCE(C(14),UT)
      EQUIVALENCE(C(2000),T)
      EQUIVALENCE(C(1666),BDIVE)
      EQUIVALENCE(C(1667),RSLANT)
      EQUIVALENCE(C(8),SPCTMO)
      IF(SPCTMO.EQ.0.0) RETURN
      UT = 0.0
      IF(AILL.EQ.0) GO TO 10
      IF(AIFAC.EQ.1) GO TO 9
      VILM = VMACH
      XIL = RSLANT * COSD(BDIVE)
      HILL= RSLANT * SIND(BDIVE)
      HILL = ABS(HILL)
      GO TO 9
   10 CONTINUE
      VILF = 0.0
    9 CONTINUE
      RILL = SQRT(XIL**2 + HILL**2)
      VSOUND = -0.00392*HILL +1117.3
      VILF = VSOUND*VILM
 1234 FORMAT(2X,9E13.4)
      WRITE(6,1234) VILM,VMACH,XIL, T ,HILL,RZE,RILL,VSOUND,VILF
      RETURN
      END
      SUBROUTINE C10
      COMMON C(4310)
      EQUIVALENCE(C(466),ZLASR),(C(467),YLASR)
      EQUIVALENCE(C(7),AILL)
      EQUIVALENCE(C(13),VILF)
      EQUIVALENCE(C(18),XIL)
      EQUIVALENCE(C(2562),HMIN)
      EQUIVALENCE(C(5),RILL)
      EQUIVALENCE(C(4),HILL)
      EQUIVALENCE(C(1),BCRE)
      EQUIVALENCE(C(2),WAND)
      EQUIVALENCE(C(3),RADIUS)
      EQUIVALENCE(C(6),AISPOT)
      EQUIVALENCE(C(9),XSPOT)
      EQUIVALENCE(C(10),YSPOT)
      EQUIVALENCE(C(14),UT)
      EQUIVALENCE(C(2000),T)
      EQUIVALENCE(C(446),CKOT)
      EQUIVALENCE(C(1651),RTXE)
      EQUIVALENCE(C(1655),RTYE)
      EQUIVALENCE(C(8),SPCTMO)
      IF(SPCTMO.EQ.0.0) RETURN
      IF(T.EQ.(HMIN/2.).AND.UT.EQ.0.0) GO TO 1
      IF(T.EQ.(HMIN/2.).AND.UT.EQ.0.1) GO TO 3
         IF(T.LT.(UT-HMIN)) GO TO 2
      IF(I.EQ.0) GO TO 4
    1 CONTINUE
      I=0
      UT = UT + CKOT
```

```
      IF(AILL.EQ.0.OR.VILF.EQ.0.0) GO TO 10
      RILL = SQRT(XIL**2+HILL**2)
   10 CONTINUE
      ANGILL = ASIND(HILL/RILL)
      BOREF = BORE*RILL/1000.0
      WANOF = WAND*RILL/1000.0
      SPWID = RADIUS*RILL/1000.0
      SPLENG = SPWID/SIND(ANGILL)
      OTWO = RONDG(BOREF)
      OTWOP = RONDG(BOREF)
      OTHREE = RONDG(WANOF)
      OTHREP = RONDG(WANOF)
      OFOUR = -SPLENG*AISPOT
      OFOURP = -SPWID*AISPOT
      OFIVE =RONDG(SPLENG*2.0)*AISPOT
      OFIVEP = RONDG(SPWID*2.0)*AISPOT
      ZLASR=OTWO+OTHREE+OFOUR+OFIVE
      YLASR=OTWOP+OTHREEP+OFOURP+OFIVEP
    3 CONTINUE
 1234 FORMAT(2X,10E13.4)
      WRITE(6,1234)XIL,VILF,HMIN,RILL,HILL,ANGILL,BOREF,BORE,WANOF,WAND
      WRITE(6,1234)SPWID,RADIUS,SPLENG,SPWID,OTWO,OTWOP,OTHREE,OTHREP,
     1OFOUR,OFOURP
      WRITE(6,1234)OFIVE,OFIVEP,XSPOT,RTXE,YSPOT,RTYE,XIL,XIL,XIL,T
      RETURN
    2 XSPOT = 0.0
      YSPOT = 0.0
      RETURN
    4 CONTINUE
      XSPOT = 0.0
      YSPOT = 0.0
      I = 1
      RETURN
      END
      FUNCTION ASIND(X)
      ASIND=57.29578*ASIN(X)
      RETURN
      END
      FUNCTION RONDG(A)
      DIMENSION X(51),Y(51)
      DATA (C=77777777777700000000B)
      DATA (Y=0.0,0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9,1.0,1.1,1.2,1.3,1.
     14,1.5,1.6,1.7,1.8,1.9,2.0,2.1,2.2,2.3,2.4,2.5,2.6,2.7,2.8,2.9,3.0,
     2 3.1,3.2,3.3,3.4,3.5,3.6,3.7,3.8,3.9,4.0,4.1,4.2,4.3,4.4,4.5,4.6,
     3 4.7,4.8,4.9,5.0 )
      DATA(X=.5,.5398,.5793,.6179,.6554,.6915,.7258,.7580,.7881,.8159,
     1.8413,.8643,.8849,.9032,.9192,.9332,.9452,.9554,.9641,.9713,.9773,
     2.9821,.9861,.9893,.9918,.9938,.9953,.9965,.9974,.9981,.9987,.9990,
     3.9993,.9995,.9997,.9998,.9998,.9999,.9999,1.,1.,1.,1.,1.,1.,1.,1.,
     41.,1.,1.,1.   )
      I = 0
      B = RANF(1.0)
      IF(B.GT..5) GO TO 1
      I = 1
      B = 1.0 - B
1     CONTINUE
      CALL CLINE(51,B,X,Y,AY)
      IF(I.EQ.1) AY = -AY
      D = AY*A
      RONDG = D .AND. C
      RETURN
      END
```

```
      SUBROUTINE CLINE(N,X,A,B,Y)
      DIMENSION A(1),B(1)
      IF(N) 100,100,101
101   DO 102 J = 2,N
      I=J
      IF(X-A(J)) 103,103,102
102   CONTINUE
103   Y = B(I-1) + (X-A(I-1)) * (B(I)-B(I-1))/(A(I)-A(I-1))
      RETURN
100   Y = 0.0
      RETURN
      END
      FUNCTION RONO(A)
      COMMONC(4310)
      EQUIVALENCE(C(2000),T)
      DATA   IE=777777777777700000000B)
      DATA   IK = 377777777777B)
      DATA(K1=37777777774008)
      IF(T.GT.0.)GOTO4
  2  IF(IX)7,8,8
  7   IX=-IX
  8   IF(IX.GT.999)GOTO3
      IX=IX*2+1
      GOTO4
  3   IX=IX/10
      GOTO2
  4   CONTINUE
      J=IX*262147
      IX=J.AND.K
      IZ=IX.AND.K1
      B=FLOAT(IZ)/3.4359739E10
      B=B*A
      RONO=B.AND.E
      RETURN
      END
      SUBROUTINE G2
      COMMON C(4310)
      DIMENSIONK(4310)
      EQIVALENCE(C(2000),T)
C**WIND AND GUSTS MODULE
C**INPUT DATA
      EQUIVALENCE(C(56),RWINC)
      EQUIVALENCE(C(58),SW1   )
      EQUIVALENCE(C(57),VWTEV)
      EQUIVALENCE(K(54),K2K   )
      EQUIVALENCE(C,K         )
      EQUIVALENCE(C(55),SW    )
      EQUIVALENCE (C(  50),OPTNW )
      EQUIVALENCE (C(  51),BPSIW )
C**OUTPUT DATA
      EQUIVALENCE (C(  52),VWTE   )
      EQUIVALENCE (C(  53),RHW    )
      EQUIVALENCE (C( 380),RANGO  )
      EQUIVALENCE (C(1317),RAIL   )
      EQUIVALENCE (C( 100),VWXE   )
      EQUIVALENCE (C( 101),VWYE   )
      EQUIVALENCE (C( 102),VWZE   )
C**INPUTS FROM OTHER MODULES
      EQUIVALENCE (C( 209),RH     )
C**CALCULATE WIND VELOCITY COMPONENTS
      IF(T.EQ.0.)OPSIWV=BPSIW
      IF(T.EQ.0.)VWTEV=VWTE
```

```
      IF(RH.GT.RHW)GOTO10
      IF (RANCC .LT. RAIL) GO TO 10
      IF(K2K.EC.1969432)COTO5
      K2K=1969432
      RANGTR=RANGO
    5 CONTINUE
      IF(RANGTR+RWINC.GT.RANGO)GOTO7
      RANGTR=RANGTR+RWINC
      CALLGAUSS(SW,DPSIW,DPSIWV)
      CALLGAUSS(SW1,VWTE,VWTEV)
    7 VWXE=-VWTEV*COSD(DPSIWV)
      VWYE=-VWTEV*SIND(DPSIWV)
      VWZE = 0.
      RETURN
   10 VWXE = 0.
      VWYE = 0.
      VWZE = 0.
      RETURN
      END
C**AIR DATA MODULE G3
      SUBROUTINE G3
      COMMON C(4310)
C**INPUT DATA
      EQUIVALENCE (C(0208),RHZRO )
C**INPUTS FROM OTHER MODULES
      EQUIVALENCE (C(0100),VWXE  )
      EQUIVALENCE (C(0101),VWYE  )
      EQUIVALENCE (C(0102),VWZE  )
      EQUIVALENCE (C(1603),VXE   )
      EQUIVALENCE (C(1607),VYE   )
      EQUIVALENCE (C(1611),VZE   )
      EQUIVALENCE (C(1623),RZE   )
C**INPUTS FROM MAIN PROGRAM
C**STATE VARIABLE OUTPUTS
C**NONE
C**OTHER OUTPUTS
      EQUIVALENCE (C(0200),VMWXE )
      EQUIVALENCE (C(0201),VMWYE )
      EQUIVALENCE (C(0202),VMWZE )
      EQUIVALENCE (C(0203),PDYNMC)
      EQUIVALENCE (C(0204),VMACH )
      EQUIVALENCE (C(0205),DRHO  )
      EQUIVALENCE (C(0206),VSOUND)
      EQUIVALENCE (C(0207),VAIRSP)
      EQUIVALENCE (C(0209),RH    )
C**CALCULATE PRESENT ALTITUDE
      RH= -RZE+RHZRO
C**CALCULATE MISSILE VELOCITY WRT AIR MASS IN EARTH AXES
      VMWXE = VXE-VWXE
      VMWYE = VYE-VWYE
      VMWZE = VZE-VWZE
      VAIRSP = SQRT(VMWXE*VMWXE+VMWYE*VMWYE+VMWZE*VMWZE)
C**AIR DENSITY, SPEED OF SOUND, DYNAMIC PRESSURE, AND MACH
      DRHO=(.076475)/(1.+.3325E-04*RH+RH*RH*RH*.02315E-12)
      VSOUND = -.00392*RH+1117.3
      PDYNMC = (DRHO*VAIRSP*VAIRSP)/64.344
      VMACH = VAIRSP/VSOUND
      RETURN
      END
      SUBROUTINE G4
C** END-OF-RUN CALCULATIONS SUBROUTINE G4
C** THIS IS A SUBROUTINE, NOT A MODULE.
```

```
C** IT IS CALLED BY STAGE 3 TO COMPUTE MISS DISTANCE AND STOP THE
C** PROGRAM IF RANGE IS ZERO.
C*********************************************************************
      COMMON C(4310)
C**INPUT DATA
C**NONE
C**INPUTS FROM OTHER MODULES
      EQUIVALENCE (C(0357),BGAMH )
      EQUIVALENCE (C(0358),BGAMV )
      EQUIVALENCE (C(0371),RANGE )
      EQUIVALENCE (C(1603),VXE    )
      EQUIVALENCE (C(1607),VYE    )
      EQUIVA ENCE (C81611),VZE    )
      EQUIVALENCE (C(1660),VTXE   )
      EQUIVA3ENCE (C(1661),VTYE   )
      EQUIVALENCE (C(.662),VTZE   )
      EQUIVA ENCE (C81615),RXE    )
      EQUIVALENCE (C(1619),RYE    )
      EQUIVALENCE (C(1623),RZE    )
      EQUIVALENCE (C(1651),RTXE   )
      EQUIVALENCE (C(1655),RTYE   )
      EQUIVALENCE (C(1659),RTZE   )
C**INPUTS FROM MAIN PROGRAM
      EQUIVALENCE (C(2000),T      )
C**STATE VARIABLE OUTPUTS
C**NONE
C**OTHER OUTPUTS
      EQUIVALENCE (C(2020),LCONV )
C** MISS DISTANCE PARAMETERS ARE OUTPUT DIRECTLY AND ARE NOT IN COMMON
C * TEST FOR INCREASING RANGE AND SOLVE FOR TIME AT WHICH RANGE IS ZERO
    5 FORMAT (1H0, 16H MISS DISTANCE= 1PE17.8/1H0,13H TIME FINAL=
     C       1PE17.8)
    6 FORMAT (1H0,10X,10HXM  EARTH=1PE17.8,3X,10HYM  EARTH=1PE17.8,3X,
     C    10HZM  EARTH=1PE17.8)
    7 FORMAT (1H0,40X,10HY FLTPATH=1PE17.8,3X 10HZ FLTPATH=1PE17.8)
C** TEST FOR INCREASING RANGE AND SOLVE FOR TIME AT WHICH RANGE IS ZERO
      IF (RANGE.GT.500.)  GO TO 20
      IF (RANGE-URANGE.LT.0.) GO TO 10
      TDEL = -(UVXT*(UXT-UXE)+UVYT*(UYT-UYE)+UVZT*(UZT-UZE)+UVXE*(UXE-
     C        UXT)+UVYE*(UYE-UYT)+UVZE*(UZE-UZT))/(UVXT*UVXT-2.*UVXT*UVXE
     C        +UVXE*UVXE+UVYT*UVYT-2.*UVYT*UVYE+UVYE*UVYE+UVZT*UVZT-2.*
     C        UVZT*UVZE+UVZE*UVZE)
      UXTO = UVXT*TDEL+UXT
      UYTO = UVYT*TDEL+UYT
      UZTO = UVZT*TDEL+UZT
      UXMO = UVXE*TDEL+UXE
      UYMO = UVYE*TDEL+UYE
      UZMO = UVZE*TDSL+UZE
      RDX = UXMO-UXTO
      RDY = UYMO-UYTO
      RDZ = UZMO-UZTO
      RMISS = SQRT(RDX*RDX+RDY*RDY+RDZ*RDZ)
      TZERO = TDEL+UT
      USTHT = SIND(BGAMV)
      UCTHT = COSD(BGAMV)
      USPSI = SIND(BGAMH)
      UCPSI = COSD (BGAMH)
      UC11 = UCTHT*UCPSI
      UC12= -USTHT*UCPSI
      UC13= USPSI
      UC21= USTHT
      UC22= UCTHT
```

```
      UC31= -UCTHT*USPSI
      UC32= USTHT*USPSI
      UC33= UCPSI
      RXFP = UC11*RDX+UC12*RDY+UC13*RDZ
      RYFP = UC21*RDX+UC22*RDY
      RZFP = UC31*RDX+UC32*RDY+UC33*RDZ
      WRITE (6,5) RMISS,TZERO
      WRITE (6,6) RDX,  RDY,  RDZ
      WRITE (6,7) RYFP,RZFP
      LCONV=2
      RETURN
   10 UT = T
      UXE = RXE
      UYE = RYE
      UZE = RZE
      UXT = RTXE
      UYT = RTYE
      UZT = RTZE
      UVXE = VXE
      UVYE = VYE
      UVZE = VZE
      UVXT = VTXE
      UVYT = VTYE
      UVZT = VTZE
   20 URANGE = RANGE
      IF (RZE .GT. 100.) LCONV = 2
      RETURN
      END
C**COORDINATE CONVERSION MODULE
      SUBROUTINE C5
      COMMON C(4310)
C
C**INPUTS FROM OTHER MODULES
      EQUIVALENCE (C(0200),VMWXE )
      EQUIVALENCE (C(0201),VMWYE )
      EQUIVALENCE (C(0202),VMWZE )
      EQUIVALENCE (C(0207),VAIRSP)
      EQUIVALENCE (C(1603),VXE    )
      EQUIVALENCE (C(1607),VYE    )
      EQUIVALENCE (C(1611),VZE    )
      EQUIVALENCE (C(1615),RXE    )
      EQUIVALENCE (C(1619),RYE    )
      EQUIVALENCE (C(1623),RZE    )
      EQUIVALENCE (C(1635),RDELX  )
      EQUIVALENCE (C(1636),RDELY  )
      EQUIVALENCE (C(1637),RDELZ  )
      EQUIVALENCE (C(1663),VDXB   )
      EQUIVALENCE (C(1664),VDYB   )
      EQUIVALENCE (C(1665),VDZB   )
      EQUIVALENCE (C(1668),RXO    )
      EQUIVALENCE (C(1669),RYO    )
      EQUIVALENCE (C(1670),RZO    )
      EQUIVALENCE (C(1703),CFA11  )
      EQUIVALENCE (C(1707),CFA12  )
      EQUIVALENCE (C(1711),CFA13  )
      EQUIVALENCE (C(1715),CFA21  )
      EQUIVALENCE (C(1719),CFA22  )
      EQUIVALENCE (C(1723),CFA23  )
      EQUIVALENCE (C(1727),CFA31  )
      EQUIVALENCE (C(1731),CFA32  )
      EQUIVALENCE (C(1735),CFA33  )
      EQUIVALENCE (C(1739),WP     )
```

```
      ECUIVALENCE (C(1743),WQ     )
      ECUIVALENCE (C(1747),WR     )
      ECUIVALENCE (C(1751),CRAD   )
      ECUIVALENCE (C(2000),T      )
      ECUIVALENCE (C(2664),DER    )
      ECUIVALENCE (C(3504),OPTN4  )
C
C**OTHER OUTPUTS
      ECUIVALENCE (C(0350),BTHT   )
      ECUIVALENCE (C(0351),BPSI   )
      ECUIVALENCE (C(0352),BPHI   )
      ECUIVALENCE (C(0353),BTHTD  )
      ECUIVALENCE (C(0354),BPSID  )
      ECUIVALENCE (C(0355),BPHID  )
      ECUIVALENCE (C(0356),VTOTC  )
      ECUIVALENCE (C(0357),BGAMH  )
      ECUIVALENCE (C(0358),BGAMV  )
      ECUIVALENCE (C(0360),VMWU   )
      ECUIVALENCE (C(0361),VMWV   )
      ECUIVALENCE (C(0362),VMWW   )
      ECUIVALENCE (C(0363),BTHLV  )
      ECUIVALENCE (C(0364),BPSLV  )
      ECUIVALENCE (C(0365),BLAMV  )
      ECUIVALENCE (C(0366),BLAMH  )
      ECUIVALENCE (C(0367),BALPHA)
      ECUIVALENCE (C(0368),BALPHY)
      ECUIVALENCE (C(0369),BALPHP)
      ECUIVALENCE (C(0370),BPHIP  )
      ECUIVALENCE (C(0371),RANGE  )
      ECUIVALENCE (C(0372),RXBA   )
      ECUIVALENCE (C(0373),RYBA   )
      ECUIVALENCE (C(0374),RZBA   )
      ECUIVALENCE (C(0375),BLOSBP)
      ECUIVALENCE (C(0376),BLOSBY)
      ECUIVALENCE (C(0377),BALPD  )
      ECUIVALENCE (C(0378),BALYD  )
      ECUIVALENCE (C(0379),BALPPC)
      ECUIVALENCE (C( 380),RANGO  )
C
C**CALCULATION OF HEADING, PITCH, ROLL EULER ANGLES IN DEGREES
      BPHI = ATAND(CFA23,CFA33)
      BTHT = ATAND(-CFA13,SQRT(CFA11*CFA11+CFA12*CFA12))
      BPSI = ATAND(CFA12,CFA11)
C
      IF(COSD(BTHT).EQ.0.0) GO TO 5
      BPSID = (WQ*SIND(BPHI)+WR*COSD(BPHI))/COSD(BTHT)
5      CONTINUE
      BPHID = WP+BPSID*SIND(BTHT)
      BTHTD = WQ*COSD(BPHI)-WR*SIND(BPHI)
C
C**CALCULATION OF TOTAL VELOCITY
      VTOTE = SQRT(VXE*VXE+VYE*VYE+VZE*VZE)
C
      RANGO = SQRT( (RXE-RXO)**2 + (RYE-RYO)**2 + (RZE-RZO)**2)
C
C**TRANSFORM MISSILE LOS FROM EARTH TO BODY AXES
      RXBA = RDELX*CFA11 + RDELY*CFA12 + RDELZ*CFA13
      RYBA = RDELX*CFA21 + RDELY*CFA22 + RDELZ*CFA23
      RZBA = RDELX*CFA31 + RDELY*CFA32 + RDELZ*CFA33
C
C**MISSILE-TGT LO  IN BODY AXES
      BLOSBP = ATAND(-RZBA,SQRT(RXBA*RXBA+RYBA*RYBA))
```

```
      BLCSBY =ATAND(RYBA,RXBA)
C
      UVP1 =  VXE*RDELX+VYE*RDELY
      UVP2 = RDELX*RDELX+RDELY*RDELY
      UVP3 = VZE*RDELZ
      UVP4 = SQRT(UVP2)
      RANGE = SQRT(UVP2+RDELZ**2)
C**VERTICAL AND HORIZONTAL LINE OF SIGHT ANGLES (EARTH AXES)
C
      BLAMH = ATAND(-RDELY,RDELX)
      BLAMV = ATAND(-RDELZ,UVP4)
C
C**VERTICAL AND HORIZONTAL PROPORTIONAL NAVIGATION ANGLES
      IF(T.LT.CER)GOTO30
      IF(RANGE.EQ.0.0) GO TO 2
      VXP=(UVP1+UVP3)/RANGE
2     CONTINUE
      IF(UVP4.EQ.0.0) GO TO 1
      VYP = ( VYE*RDELX-VXE*RDELY)/UVP4
      VZP = (VZE*UVP2-RDELZ*UVP1)/(RANGE*UVP4)
1     CONTINUE
      BTHLV  = ATAND(VZP,VXP)
      BPSLV  = ATAND(VYP,VXP)
C
      BGAMV = ATAND(-VZE,SQRT(VXE*VXE+VYE*VYE))
      BGAMH = ATAND(VYE,VXE)
C
C**VELOCITY WRT AIR IN BODY AXES
      VMWU = CFA11*VMWXE+CFA12*VMWYE+CFA13*VMWZE
      VMWV = CFA21*VMWXE+CFA22*VMWYE+CFA23*VMWZE
      VMWW = CFA31*VMWXE+CFA32*VMWYE+CFA33*VMWZE
C
C**VERTICAL AND HORIZONTAL ANGLES OF ATTACK
      BALPHA = ATAND(VMWW,VMWU)
      BALPHY = ATAND(VMWV,VMWU)
C
      USQ = VMWU**2
      IF(USQ.EQ.0.0.AND.VMWW.EQ.0.0) GO TO 3
      BALPD = (VMWU*VDZB - VMWW*VDXB)/(USQ+VMWW**2)*CRAD
3     CONTINUE
      IF(USQ.EQ.0.0.AND.VMWV.EQ.0.0) GO TO 4
      BALYD = (VMWU*VDYB - VMWV*VDXB)/(USQ+VMWV**2)*CRAD
4      CONTINUE
      BALPPD= 0.
      IF (BALPHP.GT.0.) BALPPD= (BALPHA*BALPD + BALPHY*BALYD)/BALPHP
C
C**ALPHA PRIME AND PHI PRIME (WIND TUNNEL AXES)
      IF ((BALPHA-BALPHY).EQ.0.) GO TO 30
      BPHIP=ATAND(VMWV,VMWW)
   30 BALPHP=SQRT(BALPHA**2+BALPHY**2)
      IF(ABS(BALPHP).GT.20.)BALPHP=20.
      RETURN
      END
C**   SEEKER AND PLATFORM INIT MODULE
      SUBROUTINE SII
      COMMON C(4310)
      DIMENSION IFL(100)
      EQUIVALENCE (C( 445),UT     )
      EQUIVALENCE (C( 451),CAGE   )
      EQUIVALENCE (C( 460),GUIDE  )
      EQUIVALENCE (C( 461),SAMP   )
      EQUIVALENCE (C(2541),N      )
```

```
      EQUIVALENCE (C(2562),IPL    )
      SZ=0.
      SY=0.
      UT = 0.
      GUIDE=1.
      CAGE=0.
      SAMP = 0.
      C( 452) = 0.
      IPL(N   ) = 424
      IPL(N+1) = 428
      N=N+2
      RETURN
      END
C      TIGER PLATFORM AND TRACKER MODULE
C
      SUBROUTINE S1
      COMMON C(4310)
C
C**INPUT DATA
      EQUIVALENCE (C(0441),SZGBIS)
      EQUIVALENCE (C(0442),SYGBIS)
      EQUIVALENCE (C( 443),OPTKR )
      EQUIVALENCE (C( 444),OPTBKL)
      EQUIVALENCE (C( 445),UT    )
      EQUIVALENCE (C( 446),COT   )
      EQUIVALENCE (C( 447),CKCB  )
      EQUIVALENCE (C( 448),CFOVZ )
      EQUIVALENCE (C( 449),CFOVY )
      EQUIVALENCE (C(1623),ZMSL)
      EQUIVALENCE (C( 450),GSW   )
      EQUIVALENCE (C( 451),CAGE  )
      EQUIVALENCE (C(0452),OBREAK)
      EQUIVALENCE (C(0453),RBKLOK)
      EQUIVALENCE (C(0454),DECGE )
      EQUIVALENCE(C(405),RY)
      EQUIVALENCE(C(404),RZ)
      EQUIVALENCE (C(0455),WEPSMX)
      EQUIVALENCE(C(470),SINCL)
      EQUIVALENCE(C(466),ZLASR),(C(467),YLASR)
      EQUIVALENCE(C(463),CEFOCS)
      EQUIVALENCE(C(469),CBTRAD)
      EQUIVALENCE (C(0456),CKSKR )
      EQUIVALENCE(C(473),VLAZRP)
      EQUIVALENCE(C(472),CKSK1)
      EQUIVALENCE (C(0457),CROSTP)
      EQUIVALENCE (C(0458),CROSPT)
      EQUIVALENCE (C( 460),GUIDE )
      EQUIVALENCE (C( 461),SAMP  )
      EQUIVALENCE (C( 464),CCAMVS)
      EQUIVALENCE (C( 465),CCAMHS)
      EQUIVALENCE (C(0855),GZ    )
      EQUIVALENCE (C(0056),GY    )
      EQUIVALENCE (C(3504),OPTN4 )
C
C**INPUTS FROM OTHER MODULES
      EQUIVALENCE (C(0371),RANGE )
      EQUIVALENCE (C(0372),RXBA  )
      EQUIVALENCE (C(C373),RYBA  )
      EQUIVALENCE (C(0374),RZBA  )
      EQUIVALENCE (C(1615),RXE   )
      EQUIVALENCE (C(1619),RYE   )
      EQUIVALENCE (C(1623),RZE   )
```

```
      EQUIVALENCE (C(1739),WP     )
      EQUIVALENCE (C(1740),WQD    )
      EQUIVALENCE (C(1743),WQ     )
      EQUIVALENCE (C(1744),WRC    )
      EQUIVALENCE (C(1747),WR     )
      EQUIVALENCE (C(2000),T      )
C
C**STATE VARIABLE OUTPUTS
      EQUIVALENCE (C(0424),BTHTGD)
      EQUIVALENCE (C(0427),BTHTG )
      EQUIVALENCE (C(0428),BPSIGD)
      EQUIVALENCE (C(0431),BPSIG )
C
C**OTHER OUTPUTS
      EQUIVALENCE (C(0403),EZ     )
      EQUIVALENCE (C(0407),EY     )
      EQUIVALENCE (C(0432),RXG    )
      EQUIVALENCE (C(0433),RYG    )
      EQUIVALENCE (C(0434),RZG    )
      EQUIVALENCE (C(0435),BEPSZ  )
      EQUIVALENCE (C(0436),BEPSY  )
      EQUIVALENCE (C(0437),WZ     )
      EQUIVALENCE (C(0438),WY     )
      EQUIVALENCE (C(0459),BGDEFL )
      EQUIVALENCE (C( 462),UZ     )
      EQUIVALENCE (C( 463),UY     )
C
C**DIRECTION COSINES FOR BODY TO PLATFORM TRANSFORMATION
      IF(T.GT.0.)GOTO30
      SLTS=0.
      SZ=0.
      SY-0.
   30 CONTINUE
      UB31 = SIND(BTHTG)
      UB33 = COSD(BTHTG)
      UB12 = SIND(BPSIG)
      UB22 = COSD(BPSIG)
      UB11 = UB22*UB33
      UB13 = -UB31*UB22
      UB21 = -UB33*UB12
      UB23 = UB31*UB12
      UB32 = 0.
C
C** CALCULATE TOTAL DEFLECTION OF GIMBALS
      BGDEFL=SQRT(BTHTG**2+BPSIG**2)
C
C**TRANSFORM LOS FROM BODY TO GIMBAL AXES
      RXG = UB11*RXBA+UB12*RYBA+UB13*RZBA
      RYG = UB21*RXBA+UB22*RYBA+UB23*RZBA
      RZG = UB31*RXBA+UB33*RZBA
      PC1=ATAND(-ZMSL,RANGE)
C
C**CHECK FOR MISSILE AT SEEKER BREAK-LOCK RANGE
      IF (RANGE.GT.RBKLCK) GO TO 40
      IF (CBREAK.LE.0.) GO TO 35
      IF (GICHCK.GT.0.) GO TO 49
C
C**LINE OF SIGHT RATES AFTER BREAK-LOCK
      UEPS = (T-UTIME)*WEPSMX
      IF (UEPS.GT.BEDGE) UEPS = BEDGE
      BEPSZ = UEPSZ*UEPS
      BEPSY = UEPSY*UEPS
```

```
      QICHEK = 1.
      GO TO 50
C
C**INITIALIZATION OF BREAK LOCK VARIABLES
   35 UTIME = T
      UEPSZ = BEPSZ
      UEPSY = BEPSY
      CBREAK = 1.
      WRITE (6,200) T,RANGE
  200 FORMAT (30H0BREAKLOCK HAS OCCURRED, TIME=,F8.4,8H, RANGE=,F12.4)
C
C**LOS ERRORS IN PLATFORM COORDINATES
   40 BEPSZ = ATAND(-RZG,RXG)
      BEPSY = ATAND(RYG,RXG)
      GO TO 50
   49 QICHEK = -1.
   50 CONTINUE
C
      IF (CPTKR .GT. 0.) GO TO 80
      SZ = CKSKR*BEPSZ
      SY = CKSKR*BEPSY
      CAGE = 1.
      GUIDE = 1.
      GO TO 92
   80 IF (T .LE. UT) GO TO 92
      UT = UT + CCT
      SZT=0.
      AAA=0.
      CALLGAUSS(SZT,AAA,AAV)
      IF(C(103).GT.VLAZRP)GOTO91
      SST=0.
      SSZ=ZLASR*SIND(PC1)/RANGE
      SSY=YLASR/RANGE
   60 PZG=RZG+SSZ*RANGE
      PXG=RXG
      BEPSZ=ATAND(-PZG,PXG)
      PYG=RYG+SSY*RANGE
      BEPSY=ATAND(PYG,PXG)
      CALLQUADET(CETRAD,CEFOCS,AA,BB,CC,DD,BEPSZ,BEPSY)
      CF=AA+BB+CC+DD
      IF(CF.LT.1.E-5)CF=10.
      SY= (CC-BB)*CKSKR/CF
      SZ=(CC-AA)*CKSKR/CF
      CAGE=1.
      GOTO92
   91 SZ=0.
      SY=0.
   92 UZ = SZ
      UY = SY
      IF(CF.GT.0.)CICHECK=1.
C
C**PITCH PROGRAMMING AND SEEKER GAIN SWITCHING
      IF(GUIDE.GT.0)GOTO20
      IF (SAMP  .GT. 0.) GO TO 19
      IF (CAGE .LE. 0.) GO TO 21
      UEZ = UZ
      SAMP = 1.
   19 IF (SIGN(1., UZ) .EQ. SIGN(1., UEZ)) GO TO 21
      GUIDE = 1.
   20 DAFOCS=57.6*.00667/CDT/CDT
      EZ= UZ/CKSKR*DAFOCS
      EY=UY/CKSKR*DAFOCS
```

```
      UZ=BZ
      UY=BY
      GO TO 22
   21 EZ = CGAMVS/GZ
      EY = CGAMHS/GY
      UZ = GSW*UZ
      UY = GSW*UY
   22 CONTINUE
C
C**NON-LINEAR FRICTIONAL COUPLING OF GIMBALS
      UZK=SIGN(CRCSPT,BPSICD)
      UYK=SIGN(CRCSTP,BTHTGD)
C
C**MISSILE BODY RATES IN GIMBAL AXES
      WY = UB221*WP+UB22*WQ+UB23*WR
      WZ = UB31*WP+UB32*WQ+UB33*WR
C
      BALPC = (UZ + SZGBIS + UZK)
      BETAC = (UY + SYGBIS - UYK)/UB22
C
C**GIMBAL ANGLE DERIVATIVES
      IF (CAGB .LE. 0.) GO TO 99
      BTHTGD = (BALPD - WY)/UB22
      BPSIGD = (BETAD - WZ)
C     PLG OPTION
C     BTHTGD=-CKSK3*BTHTG
C    BPSIGD=-CKSK3*BPSIG
      RETURN
   99 BTHTGD = 0.
      BPSIGD = 0.
      RETURN
      END
C**     TIGER AUTOPILOT INITIALIZATION MODULE
C******LOW  FREQUENCY MODEL******
      SUBROUTINE C1I
      COMMON C(4310)
      DIMENSION IPL(100)
      EQUIVALENCE (C( 835),EZS    )
      EQUIVALENCE (C( 843),EYS    )
      EQUIVALENCE (C( 883),EZSS   )
      EQUIVALENCE (C( 887),EYSS   )
      EQUIVALENCE (C( 464),CGAMVS)
      EQUIVALENCE (C( 465),CGAMHS)
      EQUIVALENCE (C(3504),OPTN4  )
      EQUIVALENCE (C(2561),N      )
      EQUIVALENCE (C(2562),IPL    )
C
      NHSUM = N
      IPL(N) = 800
      IPL(N+1) = 820
      IPL(N+2) = 824
      IPL(N+3) = 828
      IPL(N+4) = 832
      IPL(N+5) = 836
      IPL(N+6) = 840
      IPL(N+7) = 880
      IPL(N+8) = 884
      N = N+9
      EZS = 0.
      EYS = 0.
      GO TO 22
   21 EZS = CGAMVS
```

```
      EYS = OGAMHS
   22 EZSS = EZS
      EYSS = EYS
      C( 803) = 0.
      C( 823) = 0.
      C( 827) = 0.
      C( 831) = 0.
      C( 839) = 0.
      RETURN
      END
C**   TIGER AUTOPILOT MODULE
C*****LOW  FREQUENCY MODEL******
      SUBROUTINE C1
      COMMON C(4310)
      DIMENSION BDELTC(4),VAR(101)
C
C**INPUT DATA.
      EQUIVALENCE (C(0850),HLIMO )
      EQUIVALENCE (C(0851),HLIME )
      EQUIVALENCE (C( 852),QBIAS )
      EQUIVALENCE (C( 853),RBIAS )
      EQUIVALENCE (C(0855),GZ     )
      EQUIVALENCE (C(0856),GY     )
C     C( 857) THRU C( 860) ARE USED BY ECNTRL(I)
      EQUIVALENCE (C(0863),TAUZ   )
      EQUIVALENCE (C(0864),TAUY   )
      EQUIVALENCE (C( 865),TCY1   )
      EQUIVALENCE (C( 866),TCY2   )
      EQUIVALENCE (C( 877),TAUL   )
C
C**INPUTS FROM OTHER MODULES
      EQUIVALENCE (C(0352),BPHI   )
      EQUIVALENCE (C(0355),BPHID  )
      EQUIVALENCE (C(0403),EZ     )
      EQUIVALENCE (C(0407),EY     )
      EQUIVALENCE (C(0848),NHSUM  )
      EQUIVALENCE (C(1739),WP     )
      EQUIVALENCE (C(1740),WQD    )
      EQUIVALENCE (C(1743),WQ     )
      EQUIVALENCE (C(1744),WRD    )
      EQUIVALENCE (C(1747),WR     )
      EQUIVALENCE (C(1751),CRAD   )
C
C**INPUTS FROM MAIN PROGRAM
      EQUIVALENCE (C(2000),T      )
      EQUIVALENCE (C(2965),VAR    )
      EQUIVALENCE (C(2664),DER    )
C
C** STATE VARIABLE OUTPUTS
      EQUIVALENCE (C( 800),BPHISD)
      EQUIVALENCE (C( 803),BPHIS )
      EQUIVALENCE (C( 820),ESUMOD)
      EQUIVALENCE (C( 823),ESUMO )
      EQUIVALENCE (C( 824),ESUMED)
      EQUIVALENCE (C( 827),ESUME )
      EQUIVALENCE (C( 828),EZSDD )
      EQUIVALENCE (C( 831),EZSP  )
      EQUIVALENCE (C( 832),EZSD  )
      EQUIVALENCE (C( 835),EZS   )
      EQUIVALENCE (C( 836),EYSDD )
      EQUIVALENCE (C( 839),EYSP  )
      EQUIVALENCE (C( 840),EYSD  )
```

```
      EQUIVALENCE (C( 843),EYS    )
      EQUIVALENCE (C( 880),EZSSO )
      EQUIVALENCE (C( 883),EZSS   )
      EQUIVALENCE (C( 884),EYSSO )
      EQUIVALENCE (C( 887),EYSS   )
C
C**OUTPUTS
      EQUIVALENCE (C( 857),BDELTC)
C
C**OTHER OUTPUTS
       EQUIVALENCE(C(0876),EZRR)
C**PLATFORM RATES IN INERTIAL SPACE
      EQUIVALENCE (C(0868),EYRR   )
      EQUIVALENCE (C( 869),WQC    )
      EQUIVALENCE (C( 870),WRC    )
      EQUIVALENCE (C(0873),EODCR )
      EQUIVALENCE (C(0874),EVNCR )
      EQUIVALENCE (C(0875),BDELPC)
C
C
C**GUIDANCE SIGNAL SHAPING
      EZSO = EZSP
      EYSO = EYSP
      EZSOD = TAUZ*(TAUZ*(GZ*EZ - EZS) - 2.*EZSO)
      EYSOD = TAUY*(TAUY*(GY*EY - EYS) - 2.*EYSO)
      EZSSD = TAUZ*(EZSO/TAUL + EZS - EZSS)
      EYSSD = TAUY*(EYSO/TAUL + EYS - EYSS)
C
C**GRAVITY AND RATE BIAS
      WQC = EZSS + QBIAS
      WRC = EYSS + RBIAS
C
C**BODY RATE SHAPING AND GYRO DYNAMICS
      WQS = WQ
      WRS = WR
      IF (ABS(WQS) .GT. 30.) WQS = SIGN(30., WQS)
      IF (ABS(WRS) .GT. 30.) WRS = SIGN(30., WRS)
C
C**SUMMATION OF RATE DAMPING AND GUIDANCE SIGNALS AND THEIR DERIVATIVES
      EZRR= WQS-WQC
      EYRR = WRS - WRC
C
      UKR = .85
      IF(T.LT.TCY2)UKR=4.25
      IF(T.LT.TCY1)UKR=0.
      ESUMOD = UKR*(EZRR - EYRR)
      ESUMED = UKR*(EZRR + EYRR)
C
C**TOTAL GUIDANCE SIGNAL SHAPING AND LIMITING
      EODCR = (ESUMOD/8. + ESUMO)
      EVNCR = (ESUMED/8. + ESUME)
      IF (ABS(EODCR) .GT. HLIMO) EODCR = SIGN(HLIMO, EODCR)
      IF (ABS(EVNCR) .GT. HLIME) EVNCR = SIGN(HLIME, EVNCR)
C
C**ROLL SIGNAL SHAPING
      UKP = .33
      IF(T.LT.TCY2)UKP=1.65
      IF(T.LT.TCY1)UKP=0.
      UPHIS = UKP*(BPHID/12. + BPHI)
      BPHISD = 36.*(UPHIS - BPHIS)
      IF(ABS(BPHIS).GT.2.)BPHIS=SIGN(2.,BPHIS)
      BDELPC = -BPHIS
```

```
C
C**AUTOPILOT OUTPUT CURRENTS TO EACH ACTUATOR (FROM SUMMATION AMPS)
      BDELTC(1) = EODCR - BDELPC
      BDELTC(2) = EVNCR - BDELPC
      BDELTC(3) = EODCR + BDELPC
      BDELTC(4) = EVNCR + BDELPC
      RETURN
      END
C**    TIGER SIMPLIFIED ACTUATOR MODEL
C*******LOW  FREQUENCY MODEL******
C
      SUBROUTINE C4
C
      COMMON C(4310)
C     DIMENSION BDELTD(4),BDELT(4),BDELTC(4),VAR(101)
      DIMENSION BDELT(4),BDELTC(4)
C
C**INPUT DATA
      EQUIVALENCE(C(1121),BDMAX)
      EQUIVALENCE (C(1140),DELTPB)
      EQUIVALENCE (C(1141),DELTQB)
      EQUIVALENCE (C(1142),DELTRB)
C
C**INPUTS FROM OTHER MODULES
      EQUIVALENCE (C(1116),NHDELT)
      EQUIVALENCE (C( 857),BDELTC)
      EQUIVALENCE (C(1117),BSURF1)
      EQUIVALENCE (C(1118),BSURF2)
      EQUIVALENCE (C(1119),BSURF3)
      EQUIVALENCE (C(1120),BSURF4)
C
C**FLAP DEFLECTION BIAS
      BDELTC(1) = BDELTC(1) - DELTPB + DELTQB - DELTRB
      BDELTC(2) = BDELTC(2) - DELTPB + DELTQB + DELTRB
      BDELTC(3) = BDELTC(3) + DELTPB + DELTQB - DELTRB
      BDELTC(4) = BDELTC(4) + DELTPB + DELTQB + DELTRB
C
C**ACTUATOR DYNAMICS
      DO 30  I=1,4
      BDELT(I) = BDELTC(I)
C
C**SURFACE POSITION LIMITER
      IF(ABS(BDELT(I)).LT.BDMAX)GOTO30
      BDELT(I)=SIGN(BDMAX,BDELT(I))
   30 CONTINUE
C
      BSURF1 = BDELT(1)
      BSURF2 = BDELT(2)
      BSURF3 = BDELT(3)
      BSURF4 = BDELT(4)
C
      C(1103) = BDELT(1)
      C(1107) = BDELT(2)
      C(1111) = BDELT(3)
      C(1115) = BDELT(4)
      RETURN
      END
      SUBROUTINE A1
C
      COMMON C(4310)
C
C**TABLE LOOKUP FOR BODY FORCE COEFFICIENTS
```

```
      COMMON /NCXO /NCXO /CXOARG/ CXOA /CXOFUN/ CXOF
     1          /NCXCP / NCX /CXARG /CXA /CXFUN /CXF
     2          /NCZ /NCN /CZARG /CNA /CZFUN /CNF
     3          /NDCZ/NCDCN /DCARG/CDCNA /DCZFUN/CDCNF
     5          /NCY2 /NCY2 /CY2ARG/ CY2A /CY2FUN/ CY2F
C
C**TABLE LOOKUP FOR BODY MOMENT COEFFICIENTS
     1          /NCL2/NCL2    /CL2ARG/ CL2A     /CL2FUN/ CL2F
     2          /NCL3/ NCL3    /CL3ARG/ CL3A     /CL3FUN/ CL3F
     5          /NCM/ NCM      /CMARG / CMA      /CMFUN / CMF
     6          /NCCM/ NCDCM   /CCMARG/ CDCMA    /DCMFUN/ CDCMF
     8          /NCN2/ NCN2    /CN2ARG/ CN2A     /CN2FUN/ CN2F
     */NCLP/ NCLP    /CLPARG/ CLPA        /CLPFUN/ CLPF
     */NCMQ/ NCMQ    /CMQARG/ CMQA        /CMQFUN/ CMQF
C
C**TABLE LOOKUP FOR SURFACE COEFFICIENTS
      COMMON   /NCZD/ NCZD    /CZDARG/ CZDA    /CZDFUN/ CZDF
     2          /NCLD/ NCLD    /CLDARG/ CLDA    /CLDFUN/ CLDF
     3          /NCMD/ NCMD    /CMDARG/ CMDA    /CMDFUN/ CMDF
C
C** INPUT DATA
      EQUIVALENCE (C(1252),XINTER)
      EQUIVALENCE (C(1260),CXERR )
      EQUIVALENCE (C(1261),CZERR )
      EQUIVALENCE (C(1262),CYERR )
      EQUIVALENCE (C(1263),CLERR )
      EQUIVALENCE (C(1264),CMERR )
      EQUIVALENCE (C(1265),CNERR )
C
C**INPUTS FROM OTHER MODULES
      EQUIVALENCE (C(0204),VMACH )
      EQUIVALENCE (C(0367),BALPHA)
      EQUIVALENCE(C(2020),LCONV)
      EQUIVALENCE (C(0368),BALPHY)
      EQUIVALENCE (C(0369),BALPHP)
      EQUIVALENCE (C(0370),BPHIP )
      EQUIVALENCE (C(1117),DSURF1)
      EQUIVALENCE (C(1118),DSURF2)
      EQUIVALENCE (C(1119),BSURF3)
      EQUIVALENCE (C(1120),BSURF4)
C
C**INPUTS FROM MAIN PROGRAM
      EQUIVALENCE (C(2000),T      )
      EQUIVALENCE (C(2664),DER    )
C
C**OUTPUTS - COEFFICIENTS FOR BODY FORCES
      EQUIVALENCE (C(1203),CX     )
      EQUIVALENCE (C(1212),CXO    )
      EQUIVALENCE (C(1213),CXC    )
      EQUIVALENCE (C(1214),CNPT   )
      EQUIVALENCE (C(1215),CY2    )
      EQUIVALENCE (C(1230),BDEFL  )
      EQUIVALENCE (C(1235),CCCN   )
      EQUIVALENCE (C(1244),CNPU   )
      EQUIVALENCE (C(1245),CYPU   )
C
C**OUTPUTS - COEFFICIENTS FOR BODY MOMENTS
      EQUIVALENCE (C(1206),CLP    )
      EQUIVALENCE (C(1207),CMQ    )
      EQUIVALENCE (C(1208),CNR    )
      EQUIVALENCE (C(1209),CL     )
      EQUIVALENCE (C(1210),CM     )
```

```
      EQUIVALENCE (C(1211),CN    )
      EQUIVALENCE (C(1217),CMO   )
      EQUIVALENCE (C(1218),CN2   )
      EQUIVALENCE (C(1231),CDCN  )
      EQUIVALENCE (C(1240),CL2   )
      EQUIVALENCE (C(1241),CL3   )
      EQUIVALENCE (C(1247),CMP   )
      EQUIVALENCE (C(1248),CNP   )
      EQUIVALENCE (C(1249),CLR   )
C
C**OUTPUTS - COEFFICIENTS FOR SURFACE EFFECTS, AND TOTAL EFFECTS
      EQUIVALENCE (C(1204),CY    )
      EQUIVALENCE (C(1205),CZ    )
      EQUIVALENCE (C(1209),CL    )
      EQUIVALENCE (C(1210),CM    )
      EQUIVALENCE (C(1211),CN    )
      EQUIVALENCE (C(1219),CZQ   )
      EQUIVALENCE (C(1220),CZR   )
      EQUIVALENCE (C(1221),CMDGP )
      EQUIVALENCE (C(1222),CMR   )
      EQUIVALENCE (C(1223),CYR   )
      EQUIVALENCE (C(1224),CYQ   )
      EQUIVALENCE (C(1225),CLDRP )
      EQUIVALENCE (C(1226),CNQ   )
      EQUIVALENCE (C(1227),CLD   )
      EQUIVALENCE (C(1228),CLMP  )
      EQUIVALENCE (C(1229),CLNP  )
      EQUIVALENCE (C(1232),BCL   )
      EQUIVALENCE (C(1233),BCM   )
      EQUIVALENCE (C(1234),BCN   )
      EQUIVALENCE (C(1250),CZP   )
      EQUIVALENCE (C(1251),CYP   )
C
C  INPUT VARIABLE XINTER IS THE INTERPOLATION CONTROL
C    LESS THAN ZERO - STRAIGHT LINE INTERPOLATION
C    POSITIVE - PARABOLIC INTERPOLATION , WITH END INTERVAL
C              INTERPOLATION   (0. TO 1.)
C              0.0 - STRAIGHT LINE
C              1.0 - FULL PARABOLIC
C
C
      IF (T.LE.CER) UTIME= 0.
      IF (T-UTIME .LE. 0.) RETURN
      UTIME= T
C
C  MULTIPLE ANGLE FORMULAE AND ABSOLUTE VALUES OF ANGLE OF ATTACK
      USPHI = SIND(BPHIP)
      UCPHI = COSD(BPHIP)
      US2PHI = SIND ( 2. * BPHIP )
      US2PH2 = US2PHI ** 2
      US4PHI = SIND ( 4. * BPHIP )
      IF(ABS(BALPHY).GT.20.)BALPHY=SIGN(20.,BALPHY)
      IF(ABS(BALPHA).GT.~~.)BALPHA=SIGN(20.,BALPHA)
      UALPHA = ABS(BALPH
      UALPHY = ABS(BALPHY)
C
C**CALCULATION OF BODY FORCE COEFFICIENTS
      BDEFL = (ABS(BSURF1)+ABS(BSURF2)+ABS(BSURF3)+ABS(BSURF4))/4.
      CXO=CCCIM2(VMACH,CXOA,CXOF,NCXO,XINTER,3HCXO)
      CALL  TABL2 (BALPHP, VMACH,CXA,QXF,NCX,XINTER,3HCXC,CXC)
      IF(LCONV.EQ.2)CALLOUPT2
      IF(LCONV.EQ.2)C(2000)=C(2001)
```

```
      IF(LCONV.EQ.2)RETURN
      CALL   TABL2 (BALPHP,VMACH,CNA,CNF,NCN,XINTER,4HCNPT,CNPT)
      CALL   TABL2 (BALPHP,VMACH,COCNA,COCNF,NCOCN,XINTER,4HCOCN,COCN)
      CALL  TABL2 (BALPHP,VMACH,CY2A,CY2F,NCY2,XINTER,3HCY2,CY2)
C
      CX=CXO+CXC
      CNPU=CNPT+COCN*US2PH2
      CYPU = CY2*US4PHI
C
C  CALCULATION OF BODY MOMENT COEFFICIENTS
      CALL  TABL2 (BALPHP,VMACH,CL2A,CL2F,NCL2,XINTER,3HCL2,CL2)
      CALL  TABL2 (BALPHP,VMACH,CL3A,CL3F,NCL3,XINTER,3HCL3,CL3)
      CALL  TABL2 (BALPHP,VMACH,CMA,CMF,NCM,XINTER,3HCMO,CMO)
      CALL   TABL2 (BALPHP,VMACH,COCMA,COCMF,NCOCM,XINTER,4HCOCM,COCM)
      CALL  TABL2 (BALPHP,VMACH,CN2A,CN2F,NCN2,XINTER,3HCN2,CN2)
      CALL  TABL2 (BALPHP,VMACH,CLPA,CLPF,NCLP,XINTER,3HCLP,CLP)
      CALL  TABL2 (UALPHA,VMACH,CMQA,CMQF,NCMQ,XINTER,3HCMQ,CMQ)
      CALL  TABL2 (UALPHY,VMACH,CMQA,CMQF,NCMQ,XINTER,3HCNR,CNR)
C
      CLR = CL2*US4PHI + CL3*USPHI
      CMP = CMO+COCM*US2PH2
      CNP = CN2*US4PHI
C
C**CALCULATION OF SURFACE COEFFICIENTS
      BCL = (-BSURF1-BSURF2+BSURF3+BSURF4)/4.
      BCM = ( BSURF1+BSURF2+BSURF3+BSURF4)/4.
      BCN = (-BSURF1+BSURF2-BSURF3+BSURF4)/4.
      BCNP = -BCN
      CALL  TABL3 (BALPHP,BCM,VMACH,CZDA,CZDF,NCZD,XINTER,3HCZQ,CZQ)
      CALL  TABL3 (BALPHP,BCNP,VMACH,CZDA,CZDF,NCZD,XINTER,3HCZR,CZR)
      CALL TABL3 (BALPHP,BCM,VMACH,CMDA,CMDF,NCMD,XINTER,5HCMDQP,CMDQP)
      CALL  TABL3 (BALPHP,BCNP,VMACH,CMDA,CMDF,NCMD,XINTER,3HCMR,CMR)
      CALL  TABL3 (BALPHP,BCN,VMACH,CZDA,CZDF,NCZD,XINTER,3HCYR,CYR)
      CALL  TABL3 (BALPHP,BCM,VMACH,CZDA,CZDF,NCZD,XINTER,3HCYQ,CYQ)
      CALL TABL3 (BALPHP,BCN,VMACH,CMDA,CMDF,NCMD,XINTER,5HCLDRP,CLDRP)
      CALL  TABL3 (BALPHP,BCM,VMACH,CMDA,CMDF,NCMD,XINTER,3HCNQ,CNQ)
      CALL  TABL3 (BALPHP,BCL,VMACH,CLDA,CLDF,NCLD,XINTER,3HCLD,CLD)
C
      CNQ=-ABS(CNQ)
      CMDQP=-ABS(CMDQP)
      CLD=ABS(CLD)
      CYR=ABS(CYR)
      CYQ=ABS(CYQ)
      CZR=ABS(CZR)
      CZQ=ABS(CZQ)
      CMR=-ABS(CMR)
      CLDRP=-ABS(CLDRP)
      CZP=(CNPU+CZQ*UCPHI*BCM+CZR*USPHI*BCN)
      CYP=(CYPU+CYR*UCPHI*BCN+CYQ*USPHI*BCM)
      CL=(CLR+CLD*BCL)
      CLMP=CMP+(CMDQP*UCPHI*BCM-CMR*USPHI*BCN)
      CLNP=CNP+(CLDRP*UCPHI*BCN+CNQ*USPHI*BCM)
C
C**AERO COEFFICIENT ERRORS
      CX = CX + CXERR
      CZP = CZP + CZERR
      CYP = CYP + CYERR
      CL = CL + CLERR
      CLMP = CLMP + CMERR
      CLNP = CLNP + CNERR
C
C** TRANSFORMATION FROM WIND TO BODY AXIS
```

```
      CY = CYP*UCPHI-CZP*USPHI
      CZ = -CZP*UCPHI-CYP*USPHI
      CM = CLMP*UCPHI+CLNP*USPHI
      CN = CLNP*UCPHI-CLMP*USPHI
      RETURN
      END
C**AERO FORCE AND MOMENT MODULE   BODY AXES
      SUBROUTINE A2
      COMMON C(4310)
C
C**INPUT DATA
      EQUIVALENCE (C(1306),RFAREA)
      EQUIVALENCE (C(1307),RFLGTH)
      EQUIVALENCE (C(1308),RDELCG)
      EQUIVALENCE (C(1313),RFXCG )
      EQUIVALENCE (C(1314),RFYCG )
      EQUIVALENCE (C(1315),RFZCG )
      EQUIVALENCE (C(1316),RLLG  )
      EQUIVALENCE (C(1317),RAIL  )
      EQUIVALENCE (C(1627),AGRAV )
      EQUIVALENCE (C(3504),OPTN4 )
C
C**INPUTS FROM OTHER MODULES
      EQUIVALENCE (C(0203),PCTNHC)
      EQUIVALENCE (C(0207),VAIRSP)
      EQUIVALENCE (C( 350),BIHT  )
      EQUIVALENCE (C( 380),RANGO )
      EQUIVALENCE (C(1203),CX    )
      EQUIVALENCE (C(1204),CY    )
      EQUIVALENCE (C(1205),CZ    )
      EQUIVALENCE (C(1206),CLP   )
      EQUIVALENCE (C(1207),CMQ   )
      EQUIVALENCE (C(1208),CNR   )
      EQUIVALENCE (C(1209),CL    )
      EQUIVALENCE (C(1210),CM    )
      EQUIVALENCE (C(1211),CN    )
      EQUIVALENCE (C(1236),CM1   )
      EQUIVALENCE (C(1237),CM2   )
      EQUIVALENCE (C(1238),CM3   )
      EQUIVALENCE (C(1239),CM4   )
      EQUIVALENCE (C(1411),FTHX  )
      EQUIVALENCE (C(1412),FTHY  )
      EQUIVALENCE (C(1413),FTHZ  )
      EQUIVALENCE (C(1422),RLCG  )
      EQUIVALENCE (C(1628),DMASS )
      EQUIVALENCE (C(1723),CFA23 )
      EQUIVALENCE (C(1735),CFA33 )
      EQUIVALENCE (C(1739),WP    )
      EQUIVALENCE (C(1743),WQ    )
      EQUIVALENCE (C(1747),WR    )
      EQUIVALENCE (C(1749),FMIY  )
      EQUIVALENCE (C(1750),FMIZ  )
C
C**OTHER OUTPUTS
      EQUIVALENCE (C(1300),FXBA  )
      EQUIVALENCE (C(1301),FYBA  )
      EQUIVALENCE (C(1302),FZBA  )
      EQUIVALENCE (C(1303),FMXBA )
      EQUIVALENCE (C(1304),FMYBA )
      EQUIVALENCE (C(1305),FMZBA )
      EQUIVALENCE (C(1309),FMH1  )
      EQUIVALENCE (C(1310),FMH2  )
```

```
      EQUIVALENCE (C(1311),FMH3  )
      EQUIVALENCE (C(1312),FMH4  )
      EQUIVALENCE (C(1320),FMXTH )
      EQUIVALENCE (C(1321),FMYTH )
      EQUIVALENCE (C(1322),FMZTH )
      EQUIVALENCE (C(1323),FMXLUG)
      EQUIVALENCE (C(1324),FMYLUG)
      EQUIVALENCE (C(1325),FMZLUG)
C
C**FORCE VECTOR COMPONENTS
      UQS = PDYNMC*RFAREA
      UQSL = UQS*RFLGTH
C
      FXBA=UQS*(-CX)+FTHX
      FYBA=UQS*CY+FTHY
      FZBA=UQS*CZ+FTHZ
      IF(VAIRSP.LE.0.0) GO TO 72
C
C**AERO MOMENTS
      FMXBA = (CL+(CLP/VAIRSP)*RFLGTH*WP)*UQSL
      FMYBA = (CM+(CMQ/VAIRSP)*RFLGTH*WQ)*UQSL+FZBA*RDELCG
      FMZBA = (CN+(CNR/VAIRSP)*RFLGTH*WR)*UQSL-FYBA*RDELCG
C
C**MOMENTS CAUSED BY THRUST MISALIGNMENTS
      FMXTH = -FTHY*RFZCG + FTHZ*RFYCG
      FMYTH =  FTHX*RFZCG + FTHZ*RFXCG
      FMZTH = -FTHX*RFYCG - FTHY*RFXCG
C
C**MOMENTS AND FORCES DUE TO LUGS
      IF(RANGC.LE.RAIL+RLUG)GOTO70
      FYLUG = 0.
      FZLUG = 0.
      FMXLUG = 0.
      FMYLUG = 0.
      FMZLUG = 0.
      GO TO 74
   70 IF (RANGC .LE. RAIL) GO TO 72
      FYLUG = -(FYBA + DMASS*AGRAV*CFA23 + (FMZBA + FMZTH)*
     *             RLCG*DMASS/FMIZ)/(1. + DMASS*RLCG*RLCG/FMIZ)
      FZLUG = -(FZBA + DMASS*AGRAV*CFA33 + (FMYBA + FMYTH)*
     *             RLCG*DMASS/FMIY)/(1. + DMASS*RLCG*RLCG/FMIY)
      FMXLUG = -(FMXBA + FMXTH)
      FMYLUG = FZLUG*RLCG
      FMZLUG = FYLUG*RLCG
      GO TO 74
   72 CONTINUE
      FYLUG = -(FYBA + DMASS*AGRAV*CFA23)
      FZLUG = -(FZBA + DMASS*AGRAV*CFA33)
      FMXLUG = -(FMXBA + FMXTH)
      FMYLUG = -(FMYBA + FMYTH)
      FMZLUG = -(FMZBA + FMZTH)
   74 CONTINUE
C
C**TOTAL FORCE AND MOMENTS
      FYBA = FYBA + FYLUG
      FZBA = FZBA + FZLUG
      FMXBA = FMXBA + FMXTH + FMXLUG
      FMYBA = FMYBA + FMYTH + FMYLUG
      FMZBA = FMZBA + FMZTH + FMZLUG
C
C** CALCULATE HINGE MOMENTS
      FMH1 = CH1*UQSL
```

```
      FMH2 = CH2*UCSL
      FMH3 = CH3*UCSL
      FMH4 = CH4*UQSL
      RETURN
      END
C**INITIALIZATION FOR ENGINE MODULE
      SUBROUTINE A3I
      COMMON C(4310)
      DIMENSION IPL(101)
      EQUIVALENCE (C(2561),N      )
      EQUIVALENCE (C(2562),IPL    )
      C(1499) = 0.
      IPL(N  ) = 1496
      N = N+1
      RETURN
      END
C**ENGINE MODULE
      SUBROUTINE A3
      COMMON C(4310)
C
C**LOOK UP TABLE FOR THRUST
      COMMON  /NTH/NTH   /THARG/THA   /THFUN/THF
C
C** INPUT DATA
      EQUIVALENCE (C(1401),BALPHT)
      EQUIVALENCE (C(1402),BPHIT )
      EQUIVALENCE (C(1403),CNALGN)
      EQUIVALENCE (C(1404),PCFTM )
      EQUIVALENCE (C(1405),QBURN )
      EQUIVALENCE (C(1414),CISP  )
      EQUIVALENCE (C(1415),CWT   )
      EQUIVALENCE (C(1416),DMP   )
      EQUIVALENCE (C(1417),RCCGO )
      EQUIVALENCE (C(1418),RCCGF )
      EQUIVALENCE (C(1419),FMIXO )
      EQUIVALENCE (C(1420),FMIYO )
      EQUIVALENCE (C(1421),RLCGO )
      EQUIVALENCE (C(1627),AGRAV )
C
C** INPUTS FROM OTHER MODULES
      EQUIVALENCE (C(1252),XINTER)
      EQUIVALENCE (C(2000),T     )
C
C** OUTPUTS
      EQUIVALENCE (C(1308),RDELCG)
      EQUIVALENCE (C(1409),UCMP  )
      EQUIVALENCE (C(1410),FTHRST)
      EQUIVALENCE (C(1411),FTHX  )
      EQUIVALENCE (C(1412),FTHY  )
      EQUIVALENCE (C(1413),FTHZ  )
      EQUIVALENCE (C(1422),RLCG  )
      EQUIVALENCE (C(1628),DMASS )
      EQUIVALENCE (C(1748),FMIX  )
      EQUIVALENCE (C(1749),FMIY  )
      EQUIVALENCE (C(1750),FMIZ  )
C
C**STATE VARIABLES AND THEIR DERIVATIVES
      EQUIVALENCE (C(1496),UIMPD )
      EQUIVALENCE (C(1499),UIMP  )
C
      IF (QBURN.GT.0.) RETURN
      FTHRST=CCOIN2(T,THA,THF,NTH,XINTER,8HFTHRST)
```

```
      FTHRST = FTHRST*(1. + PCFTH)
C
   10 USINA=SIND(BALPHT)
      FTHX=FTHRST*COSD(BALPHT)
      FTHY=-FTHRST*USINA*SIND(BPHIT)
      FTHZ=FTHRST*USINA*COSD(BPHIT)
      GO TO 30
   20 FTHX=FTHRST
      FTHY=0.
      FTHZ=0.
   30 CONTINUE
C
      UIMPD = FTHRST
      UDWP = UIMP/CISP
C
      DMASS = (DWT - UDWP)/AGRAV
      RDELCG = RDCGO - (RDCGO - RDCGF)*UDWP/DWP
C
      FMIX = FMIXO*(DWT - UDWP)/DWT
      FMIY = FMIYO*(DWT - UDWP)/DWT
      FMIZ = FMIY
      RLCG = RLCGO + RDELCG
      IF (FTHRST .GT. 0.) RETURN
C
      WRITE (6,100) T
  100 FORMAT (//14H BURNOUT TIME=,F8.4,5H SEC.)
      CBURN=1.0
      FTHRST=0.
      FTHX=0.
      FTHY=0.
      FTHZ=0.
      RETURN
      END
C**   TRANSLATIONAL DYNAMICS INITIALIZATION MODULE FOR D1
      SUBROUTINE D1I
      COMMON C(4310)
      EQUIVALENCE (C(2561),N      )
      EQUIVALENCE (C(2562),IPL    )
      DIMENSION  IPL (100)
C
C** INPUT DATA
      EQUIVALENCE (C( 100),VWXE  )
      EQUIVALENCE (C( 101),VWYE  )
      EQUIVALENCE (C( 102),VWZE  )
      EQUIVALENCE (C( 204),VMACH )
      EQUIVALENCE (C( 208),RHZRO )
      EQUIVALENCE (C( 367),DALPHA)
      EQUIVALENCE(C(371),RANGE)
      EQUIVALENCE(C(1752),BPHIO)
      EQUIVALENCE (C(3501),EMTARG)
      EQUIVALENCE (C( 368),BALPHY)
      EQUIVALENCE (C( 427),BTHTG )
      EQUIVALENCE (C( 431),BPSIG )
      EQUIVALENCE (C(1405),CBURN )
      EQUIVALENCE (C(1639),OPTARG)
      EQUIVALENCE (C(1666),BDIVE )
      EQUIVALENCE (C(1667),RSLANT)
      EQUIVALENCE (C(1739),WP     )
      EQUIVALENCE (C(1743),WQ     )
      EQUIVALENCE (C(1747),WR     )
      EQUIVALENCE (C(3502),OPTN2 )
      EQUIVALENCE (C(3504),OPTN4 )
```

```
C
C** OUTPUT TO MODULES
      EQUIVALENCE (C( 370),BPHIP )
      EQUIVALENCE (C(1615),RXE   )
      EQUIVALENCE (C(1619),RYE   )
      EQUIVALENCE (C(1623),RZE   )
      EQUIVALENCE (C(1603),VXE   )
      EQUIVALENCE (C(1607),VYE   )
      EQUIVALENCE (C(1611),VZE   )
      EQUIVALENCE (C(1629),ATHRST)
      EQUIVALENCE (C(1635),RDELX )
      EQUIVALENCE (C(1636),RDELY )
      EQUIVALENCE (C(1637),RDELZ )
      EQUIVALENCE(C(1644),ATARG)
      EQUIVALENCE(C(1647),VTARG)
      EQUIVALENCE (C(1651),RTXE  )
      EQUIVALENCE (C(1655),RTYE  )
      EQUIVALENCE (C(1659),RTZE  )
      EQUIVALENCE (C(1668),RXO   )
      EQUIVALENCE (C(1669),RYO   )
      EQUIVALENCE (C(1670),RZO   )
      EQUIVALENCE (C(1753),BTHTO )
      EQUIVALENCE (C(1754),BPSIO )
C
C
      C(1647)=C(1648)
      IPL(N) = 1600
      IPL(N+1) = 1604
      IPL(N+2) = 1608
      IPL(N+3) = 1612
      IPL(N+4) = 1616
      IPL(N+5) = 1620
      IPL(N+6) = 1640
      IPL(N+7) = 1644
      IPL(N+8) = 1648
      IPL(N+9) = 1652
      IPL(N+10) = 1656
      IPL(N+11)=1672
      N=N+12
      IF(EMTARG)10,9,10
   10 ATARG=ATHRST/EMTARG
      GOTO11
    9 ATARG=0.
   11 CONTINUE
      BPHIO=0.
C**CALCULATE MISSILE PARAMETER INITIAL CONDITIONS
      BTHTO=BDIVE+BALPHA
      BPSIO=-BALPHY
      RXE=-RSLANT*COSD(BDIVE)
      RANGE=RSLANT
      RYE=RSLANT*SIND(BALPHY)
      RZE= RSLANT*SIND(BDIVE)
   20 RH = RHZRO - RZE
      C(427)=-BALPHA
      C(431)=BALPHY
      C(431)=-BALPHY
C
      USTHT = SIND(BTHTO)
      UCTHT = COSD(BTHTO)
      UCPSI = COSD(BPSIO)
      USPSI = SIND(BPSIO)
C
```

```
      RXBA = -UCPSI*UCTHT*RXE + USTHT*RZE
      RYBA =         USPSI*RXE
      RZBA = -UCPSI*USTHT*RXE - UCTHT*RZE
C
   24 VSOUND = 1117.3 - .00392*RH
      VMWTE = VMACH*VSOUND
      VMWXY = VMWTE*COSD(BALPHA - BTHTO)
      VXE=VWXE+VMWXY*COSD(BALPHY)
      VYE=VWYE-VMWXY*SIND(BALPHY)
      VZE=VWZE-VMWTE*SIND(BDIVE)
C
   30 RDELX = RTXE-RXE
      RDELY = RTYE-RYE
      RDELZ = RTZE-RZE
      RXO = RXE
      RYO = RYE
      RZO = RZE
      RETURN
      END
C**TRANSLATIONAL DYNAMICS MODULE
      SUBROUTINE D1
      COMMON C(4310)
C
C**INPUT DATA
      EQUIVALENCE (C(1421),RAIL   )
      EQUIVALENCE (C(1627),AGRAV  )
      EQUIVALENCE (C(1628),CMASS  )
      EQUIVALENCE (C(1629),ATHRST )
      EQUIVALENCE (C(1630),ATURNT )
      EQUIVALENCE (C(1631),BCAMT  )
      EQUIVALENCE (C(1680),OPTNO  )
      EQUIVALENCE (C(1681),ADIVE  )
      EQUIVALENCE (C(1751),CRAD   )
      EQUIVALENCE (C(3504),OPTN4  )
C
C**INPUTS FROM OTHER MODULES
      EQUIVALENCE (C( 371),RANGE  )
      EQUIVALENCE (C( 380),RANGO  )
      EQUIVALENCE (C(1300),FXBA   )
      EQUIVALENCE (C(1301),FYBA   )
      EQUIVALENCE (C(1302),FZBA   )
      EQUIVALENCE (C(1667),RZ     )
      EQUIVALENCE (C(1703),CFA11  )
      EQUIVALENCE (C(1707),CFA12  )
      EQUIVALENCE (C(1711),CFA13  )
      EQUIVALENCE (C(1715),CFA21  )
      EQUIVALENCE (C(1719),CFA22  )
      EQUIVALENCE (C(1723),CFA23  )
      EQUIVALENCE (C(1727),CFA31  )
      EQUIVALENCE (C(1731),CFA32  )
      EQUIVALENCE (C(1735),CFA33  )
      EQUIVALENCE (C(2000),T      )
C
C**STATE VARIABLE OUTPUTS
      EQUIVALENCE (C(1600),VXED   )
      EQUIVALENCE (C(1603),VXE    )
      EQUIVALENCE (C(1604),VYED   )
      EQUIVALENCE (C(1607),VYE    )
      EQUIVALENCE (C(1608),VZED   )
      EQUIVALENCE (C(1611),VZE    )
      EQUIVALENCE (C(1612),RXED   )
      EQUIVALENCE (C(1615),RXE    )
```

```
      ECUIVALENCE (C(1616),RYED   )
      ECUIVALENCE (C(1619),RYE    )
      ECUIVALENCE (C(1620),RZED   )
      ECUIVALENCE (C(1623),RZE    )
      ECUIVALENCE(C(1644),ATARG)
      ECUIVALENCE(C(1647),VTARG)
      ECUIVALENCE(C(1672),B..TD)
      ECUIVALENCE(C(1675),BPSIT )
      ECUIVALENCE  (C(3501),FNTARG)
      ECUIVALENCE (C(1648),RTXED )
      ECUIVALENCE (C(1651),RTXE  )
      ECUIVALENCE (C(1652),RTYED )
      ECUIVALENCE (C(1655),RTYE  )
      ECUIVALENCE (C(1656),RTZED )
      ECUIVALENCE (C(1659),RTZE  )
C
C**OTHER CLTPUTS
      ECUIVALENCE (C(1624),AXBA  )
      ECUIVALENCE (C(1625),AYBA  )
      ECUIVALENCE (C(1626),AZBA  )
      ECUIVALENCE (C(1632),VCELX )
      ECUIVALENCE (C(1633),VCELY )
      ECUIVALENCE (C(1634),VCELZ )
      ECUIVALENCE (C(1635),RCELX )
      ECUIVALENCE (C(1636),RCELY )
      ECUIVALENCE (C(1637),RCELZ )
      ECUIVALENCE (C(1630),VCLSNG)
      ECUIVALENCE (C(1660),VTXE  )
      ECUIVALENCE (C(1661),VTYE  )
      ECUIVALENCE (C(1662),VTZE  )
      ECUIVALENCE (C(1663),VOXB  )
      ECUIVALENCE (C(1664),VOYB  )
      ECUIVALENCE (C(1665),VOZB  )
      ECUIVALENCE (C(1676),ANGX  )
      ECUIVALENCE (C(1677),ANGY  )
      ECUIVALENCE (C(1678),ANGZ  )
C
      C(1647)=C(1648)
C**ADD AERO AND THRUST FORCES TO GET TOTAL ACCELERATION IN BODY AXES
      AXBA = FXBA/CMASS
      AYBA = FYBA/CMASS
      AZBA = FZBA/CMASS
C
C**RESCLVE FRCM BCCY TO EARTH AXES
      AXE = CFA11*AXBA+CFA21*AYBA+CFA31*AZBA
      AYE = CFA12*AXBA+CFA22*AYBA+CFA32*AZBA
      AZE = CFA13*AXBA+CFA23*AYBA+CFA33*AZBA
C
C**INTEGRATE ACCELERATIONS
      VXEC = AXE
      VYEC = AYE
      VZEC = AZE + AGRAV
C
C** CALCULATE TOTAL MISSILE ACCELERATION IN BODY AXES
      VOXB = CFA11*VXED + CFA12*VYED + CFA13*VZED
      VOYB = CFA21*VXEC + CFA22*VYED + CFA23*VZED
      VCZB = CFA31*VXEC + CFA32*VYED + CFA33*VZED
      IF (AGRAV.LE.0.) GO TO 10
      ANGX = VOXB/AGRAV
      ANGY = VCYB/AGRAV
      ANGZ = VOZB/AGRAV
C
```

```
C**INTEGRATE VELOCITIES TO EARTH AXES  POSITION
      RXED=VXE
      RYED = VYE
      RZED = VZE
      IF(EMTARG)10,9,10
   10 ATARG=ATHRST/EMTARG
      GOTO11
    9 ATARG=0.
   11 CONTINUE
      BPSITC= 0.
      IF (VTARG.GT.0.) BPSITC= ATURNT*AGRAV*CRAD/VTARG
C
      RTXED = VTXE
      RTYED = VTYE
      RTZED = VTZE
C
      VDELX = VTXE-VXE
      VDELY = VTYE-VYE
      VDELZ = VTZE-VZE
C
      RDELX = RTXE-RXE
      RDELY = RTYE-RYE
      RDELZ = RTZE-RZE
      VCLSNG = (RDELX*VDELX+RDELY*VDELY+RDELZ*VDELZ)/RANGE
      RETURN
      END
C**ROTATIONAL DYNAMICS INITIALIZATION MODULE D2IEUL
      SUBROUTINE D2I
      COMMON C(4310)
      DIMENSION  IPL (100)
C**NONE
C**INPUTS FROM MAIN PROGRAM
      EQUIVALENCE (C(2561),N        )
      EQUIVALENCE (C(2562),IPL      )
C**STATE VARIABLE OUTPUTS
      EQUIVALENCE (C(1703),CFA11 )
      EQUIVALENCE (C(1707),CFA12 )
      EQUIVALENCE (C(1711),CFA13 )
      EQUIVALENCE (C(1715),CFA21 )
      EQUIVALENCE (C(1719),CFA22 )
      EQUIVALENCE (C(1723),CFA23 )
      EQUIVALENCE (C(1727),CFA31 )
      EQUIVALENCE (C(1731),CFA32 )
      EQUIVALENCE (C(1735),CFA33 )
C**OTHER OUTPUTS
C**NONE
C**INITIAL CALCULATION OF EULER ANGLE MATRIX OF DIRECTION COSINES (CFA)
      USPHI = SIND(BPHIO)
      UCPHI = COSD(BPHIO)
      USTHT = SIND(BTHTO)
      UCTHT = COSD(BTHTO)
      USPSI = SIND(BPSIO)
      UCPSI = COSD(BPSIO)
      CFA11 = UCPSI*UCTHT
      CFA12 = USPSI*UCTHT
      CFA13 = -USTHT
      CFA21 = -USPSI*UCPHI+UCPSI*USTHT*USPHI
      CFA22 = UCPSI*UCPHI+USPSI*USTHT*USPHI
      CFA23 = UCTHT*USPHI
      CFA31 = UCPSI*USTHT*UCPHI+USPSI*USPHI
      CFA32 = USPSI*USTHT*UCPHI-UCPSI*USPHI
      CFA33 = UCTHT*UCPHI
```

```
C
C**INTEGRATEC PARAMATER LIST (IPL) FOR WPD,WQD,WRD,AND CFAD
      IPL(N) = 1700
      IPL(N+ ) = 1704
      IPL(N+2) = 1708
      IPL(N+3) = 1712
      IPL(N+4) = 1716
      IPL(N+5) = 1720
      IPL(N+6) = 1724
      IPL(N+7) = 1728
      IPL(N+8) = 1732
      IPL(N+9) = 1736
      IPL(N+10) = 1740
      IPL(N+11) = 1744
      N = N+12
C** RESET ANGULAR RATE DERIVATIVES TO ZERO.
      C(1700) = 0.
      C(1704) = 0.
      C(1708) = 0.
      C(1712) = 0.
      C(1716) = 0.
      C(1720) = 0.
      C(1724) = 0.
      C(1728) = 0.
      C(1732' = 0.
      C(1736) = 0.
      C(1740) = 0.
      C(1744) = 0.
      RETURN
      END
C**   RCTATICNAL CYNAMICS MODULE
      SUBROUTINE C7
      COMMON C(4310)
C
C**DATA INPUTS
      EQUIVALENCE (C(1421),RAIL  )
      EQUIVALENCE (C(1748),FMIX  )
      EQUIVALENCE (C(1749),FMIY  )
      EQUIVALENCE (C(1750),FMIZ )
      EQUIVALENCE (C(1751),CRAO  )
      EQUIVALENCE (C(3503),OPTN3)
      EQUIVALENCE (C(3504),OPTN4 )
C
C**INPUTS FROM OTHER MODULES
      EQUIVALENCE (C( 380),RANGO )
      EQUIVALENCE (C(1303),FMXBA )
      EQUIVALENCE (C(1304),FMYBA )
      EQUIVALENCE (C(1305),FMZBA )
      EQUIVALENCE (C(1308),RCELCG)
C
C *INPUTS FROM MAIN PROGRAM
C**STATE VARIABLE OUTPUTS
      EQUIVALENCE (C(1700),CFA11C)
      EQUIVALENCE (C(1703),CFA11 )
      EQUIVALENCE (C(1704),CFA12C)
      EQUIVALENCE (C(1707),CFA12 )
      EQUIVALENCE (C(1708),CFA13C)
      EQUIVALENCE (C(1711),CFA13 )
      EQUIVALENCE (C(1712),CFA21C)
      EQUIVALENCE (C(1715),CFA21 )
      EQUIVALENCE (C(1716),CFA22C)
      EQUIVALENCE (C(1719),CFA22 )
```

```
      EQUIVALENCE (C(1720),CFA23D)
      EQUIVALENCE (C(1723),CFA23 )
      EQUIVALENCE (C(1724),CFA31D)
      EQUIVALENCE (C(1727),CFA31 )
      EQUIVALENCE (C(1728),CFA32D)
      EQUIVALENCE (C(1731),CFA32 )
      EQUIVALENCE (C(1732),CFA33D)
      EQUIVALENCE (C(1735),CFA33 )
      EQUIVALENCE (C(1736),WPD    )
      EQUIVALENCE (C(1739),WP     )
      EQUIVALENCE (C(1740),WQD    )
      EQUIVALENCE (C(1743),WQ     )
      EQUIVALENCE (C(1744),WRD    )
      EQUIVALENCE (C(1747),WR     )
C
C**INTEGRATE BODY ANGULAR RATES
      WPD = CRAD*FMXBA/FMIX
   55 WQD = (CRAD*FMYBA+(FMIZ-FMIX)*WP*WR/CRAD)/FMIY
   65 WRD = (CRAD*FMZBA+(FMIX-FMIY)*WP*WQ/CRAD)/FMIZ
C
C**INTEGRATE ATTITUDE DIRECTION COSINES
   49 CFA11D=(CFA21*WR-CFA31*WQ)/CRAD
      CFA12D=(CFA22*WR-CFA32*WQ)/CRAD
      CFA13D=(CFA23*WR-CFA33*WQ)/CRAD
      CFA21D = (CFA31*WP-CFA11*WR)/CRAD
      CFA22D = (CFA32*WP-CFA12*WR)/CRAD
      CFA23D = (CFA33*WP-CFA13*WR)/CRAD
      CFA31D = (CFA11*WQ-CFA21*WP)/CRAD
      CFA32D = (CFA12*WQ-CFA22*WP)/CRAD
      CFA33D = (CFA13*WQ-CFA23*WP)/CRAD
      RETURN
      END
C     BASIC INPUT SUBROUTINE DINPT1
      SUBROUTINE DINPT1
      COMMON C(4310)
      EQUIVALENCE(C(2800),B)
      EQUIVALENCE(K(4310),KK)
      EQUIVALENCE (C(3218),ONAME1), (C(3268),ONAME2), (C(3318),ONAME3),
     C            (C(3328),ONAME4), (C(2361),NOMOD ), (C(2362),MODNO ),
     C            (C(3440),NORNOM), (C(3441),RNDMNO), (C(3167),NCOUT ),
     C            (C(3168),OUTNO ), (C(2461),NOSUB ), (C(2462),SUBNO ),
     C            (C(2355),IR    ), (C(2357),VR    ), (C(3339),NOSTAT),
     C            (C(3338),LOSTAT), (C(3340),STATNO), (C(3066),NOLIST),
     C            (C(3067),LISTNO), (C(3117),VALUE ), (C(2008),PLOTNO),
     C            (C(2009),NOPLOT), (C(2325),VLABLE), (C,K)
      EQUIVALENCE (C(1984),NPLOT )
      EQUIVALENCE (C(1985),OUTPLT)
      DOUBLE PRECISION ALPHA
      DIMENSION ONAME3(10),ONAME4(10)
      DIMENSION LISTNO(50), VALUE(50)
      DIMENSION SUBNO(99),IR(2),VR(2)
      DIMENSION RNDMNO(50)
      DIMENSION B(100)
      DIMENSION ALPHA(3),ONAME1(50),ONAME2(50),OUTNO(50) ,MODNO(99)
      DIMENSION K(4310)
      DIMENSION STATNO (100)
      DIMENSION VLABLE(2,15)
      DIMENSION OUTPLT(15)
      REAL MODNO
      INTEGER OUTNO
      INTEGER RNDMNO
      INTEGER OUTPLT
```

```
      INTEGER STATNO
      IF(K(4310))56,55,56
55    KK=0
56    CONTINUE
      JAR = 0
      WRITE(6,31)
   31 FORMAT(11H1INPUT DATA/)
    1 READ(5,2) IR(1),ALPHA(1),ALPHA(2),ALPHA(3),IR(2),VR(1),VR(2)
      WRITE(6,30)IR(1),ALPHA(1),ALPHA(2),ALPHA(3),IR(2),VR(1),VR(2)
   30 FORMAT(I2,3A6,I5,5X,1P2E15.7)
    2 FORMAT(I2,3A6,I5,5X,2E15.9)
    7 IF( IR(1) .NE. 1 )  GO TO 3
      NOSUB = NOSUB + 1
      SUBNO(NOSUB) = IR(2)
      K1=KK+3510
      C(K1)=IR(2)
      K(K1+1)=2461+NOSUB
      KK=KK+2
      GO TO 1
    3 IF( IR(1) .NE. 2 )  GO TO 4
      NOMOD = NOMOD + 1
      MODNO(NOMOD) = IR(2)
      K1=KK+3510
      C(K1)=IR(2)
      K(K1+1)=2361+NOMOD
      KK=KK+2
      GO TO 1
    4 IF(IR(1) .NE. 3) GO TO 5
      L = IR(2)
      C(L) = VR(1)
      K1=KK+3510
      C(K1)=VR(1)
      K(K1+1)=IR(2)
      KK=KK+2
      IF (VR(2) .EQ. 0.) GO TO 1
      NOLIST = NOLIST + 1
      LISTNO(NOLIST) = L
      VALUE(NOLIST) = VR(1)
      K1=KK+3510
      K(K1)=L
      K(K1+1)=3066+NOLIST
      C(K1+2)=VR(1)
      K(K1+3)=3116+NOLIST
      KK=KK+4
      GO TO 1
    5 IF(IR(1) .NE. 4)GO TO 6
      NOOUT = NOOUT + 1
      IF (NOOUT.GT.50) GO TO 1
      ONAME1(NOOUT)=ALPHA(2)
      ONAME2(NOOUT) = ALPHA(3)
      OUTNO(NOOUT) = IR(2)
      K1=KK+3510
      C(K1)=ALPHA(2)
      K(K1+1)=3217+NOOUT
      C(K1+2)=ALPHA(3)
      K(K1+3)=3267+NOOUT
      K(K1+4)=IR(2)
      K(K1+5)=3167+NOOUT
      KK=KK+6
      GO TO 1
    6 IF (IR(1) .NE. 5) GO TO 16
      IF (VR(1) .EQ. 0.) GO TO 17
```

```
      LCSTAT = LOSTAT + 1
   17 NCSTAT = NCSTAT + 1
      STATNC(NCSTAT) = IR(2)
      CNAME3(NCSTAT)=ALPHA(2)
      CNAME4(NCSTAT)=ALPHA(3)
      GO TO 1
   16 IF (IR(1).NE.7) GO TO 19
      NPLOT=NPLOT+1
      IF (NPLOT.GT.15) GO TO 1
      DO 20 I=1,2
   20 VLABLE (I,NPLOT)=ALPHA(I+1)
      OUTPLT(NPLOT)=IR(2)
      GO TO 1
 19   CONTINUE
      K(4309)=NCSUB
      K(4308)=NCMCC
      K(4307)=NCLIST
      K(4306)=NCOUT
      K(4305)=LCSTAT
      K(4304)=NCSTAT
      K(4303)=NPLOT
      K(4302)=NORNCM
      IF(IR(2).EQ.0)RETURN
      N = IR(2)
      DO 12 I = 1, N
      READ (5,13) J,Y,MANO,MIER,SIGNO,BETA
      WRITE (6,13) J,Y,MANO,MIER,SIGNO,BETA
   13 FORMAT (I5,F5.0,2(3X,I12),2E15.8)
      NORNCM = NORNCM + 1
      RNCMNC(I) = J
      C(J) = Y
      K(J+1) = MANO
      K(J+2) = MIER
      C(J+3) = SIGNO
      K1=KK+3510
      K(K1)=MANO
      K(K1+1)=J+1
      K(K1+2)=MIER
      K(K1+3)=J+2
      C(K1+4)=SIGNO
      K(K1+5)=J+3
      C(K1+6)=Y
      K(K1+7)=J
      C(K1+8)=J
      K(K1+9)=3440+I
      KK=KK+10
 12   CONTINUE
      RETURN
      END
C     OUTPUT INITIALIZITION SUBROUTINE OUPT2
      SUBROUTINE OUPT2
      COMMON C(4310),GRAPH
      EQUIVALENCE (C(2017),DTCNT ), (C(3167),NOOUT ), (C(2016),PGCNT ),
     C            (C(2014),ITCNT ), (C(2003),PCNT  ), (C(2015),CPP   ),
     C            (C(2018),TAPE  ), (C(2019),TAPEND), (C(2013),OCC   ),
     C            (C(2000),T     ), (C(2021),KCONV ), (C(2025),TIME  ),
     C            (C(2009),PLOTNC), (C(2009),NOPLOT), (C(3168),OUTNO ),
     C            (C(2004),PPNT  ), (C(2023),OPOINT)
      DIMENSION GRAPH(2,2),TIME(2),OUTNO(50)
      INTEGER     PGCNT , DTCNT , OUTNO , OPOINT
      EQUIVALENCE (C(1985),OUTPLT)
      INTEGER OUTPLT
```

```
      DIMENSION OUTPLT(15)
      KCONV=0
      ITCNT = DDD + 1.0
      PCNT = T-0.000001
      PPNT=PCNT
      PCCNT = 1
      DTCNT = (NOOUT + 4)/5
      IF ( ITCNT .GE. 7) GO TO 2
      WRITE(6,6)(I,C(I),C(I+1),C(I+2),C(I+3),C(I+4),C(I+5),C(I+6),I=1,35
     *10,7)
    6 FORMAT(1H1/(I5,1P7E15.7))
C
    2 TIME(1)=T
      CPOINT =1
      DO 10 J=1,NOPLOT
      K=OUTPLT(J)
   10 GRAPH(1,J)=C(K)
      RETURN
      END
C     OUTPUT SUBROUTINE OUPT3
      SUBROUTINE OUPT3
      COMMON C(4310),GRAPH
      EQUIVALENCE (C(3168),OUTNO ), (C(3218),ONAME1), (C(3266),ONAME2),
     C            (C(2017),DTCNT ), (C(3167),NOOUT ), (C(2016),PCCNT ),
     C            (C(2014),ITCNT ), (C(2003),PCNT  ), (C(2015),CPP   ),
     C            (C(2000),T     ), (C(2664),DER   ), (C(2018),TAPE  ),
     C            (C(2019),TAPEND), (C(2008),PLOTNO), (C(2009),NOPLOT),
     C            (C(2005),PPP   ), (C(2004),PPNT  ), (C(2025),TIME  ),
     C            (C(2023),CPOINT)
      EQUIVALENCE (C(1985),OUTPLT)
      DIMENSION B(50),OUTNO(50),ONAME1(50),ONAME2(50)
      DIMENSION GRAPH(2,2),TIME(2)
      DIMENSION OUTPLT(15)
      INTEGER DTCNT,PCCNT,OUTNO
      INTEGER CPOINT
      INTEGER OUTPLT
      IF (ITCNT. GT. 6) GO TO 7
      ITCNT = ITCNT + 1
C
      WRITE(6,6)(I,C(I),C(I+1),C(I+2),C(I+3),C(I+4),C(I+5),C(I+6),I=1,35
     *10,7)
    6 FORMAT(1H1/(I5,1P7E15.7))
      PCCNT = 1
    7 IF (DER. EQ. DER1) GO TO 8
      DER1 = DER
      WRITE(6,20)T,DER
   20 FORMAT(1H ,5HTIME=F14.7,2X,10HSTEP SIZE=1PE19.7)
    8 IF (T .LT. PCNT)GOTO15
    9 PCNT = PCNT + CPP
      IF (PCCNT. NE. 1) GO TO 3
    1 WRITE(6,2) (ONAME1(I),ONAME2(I), I=1,NOOUT)
    2 FORMAT (1H1,3X,4HTIME,5X,5(7X,2A6)//(20X,2A6,7X,2A6,7X,2A6,7X,
     12A6,7X,2A6)/)
      PCCNT = 2*DTCNT + 4
    3 IF(PCCNT .GE. 86) GO TO 1
      DO 4 I = 1,NOOUT
      J = OUTNO(I)
    4 B(I) = C(J)
      WRITE (6,5) T,(B(I), I = 1,NOOUT)
    5 FORMAT (///,F14.7,1P5E19.7/(14X,1P5E19.7))
      PCCNT = PCCNT + DTCNT + 4
   15 IF(T.LT.PPNT.OR.NOPLOT.EQ.0)RETURN
```

```
      PPNT=PPNT+PPP
      KPOINT =CPOINT +1
      IF (KPOINT-300) 16,13,18
   13 WRITE (6,14)
   14 FORMAT (///1H **** WARNING-PLOTTING ARRAY FILLED-ONLY FIRST 300 P
     COINTS PLOTTED ****,//)
   16 CPOINT=KPOINT
      TIME (CPOINT)=T
      DO 10 J=1,NOPLOT
      M=OUTPLT(J)
   10 GRAPH(CPOINT   ,J)=C(M)
   18 RETURN
      END
      SUBROUTINE ZERO
      COMMON C(4310)
      EQUIVALENCE (C(1984),NPLOT )
      EQUIVALENCE (C(2023),OPOINT)
      EQUIVALENCE (C(2361),NCMOD )
      EQUIVALENCE (C(2461),NOSUB )
      EQUIVALENCE (C(3066),NOLIST)
      EQUIVALENCE (C(3167),NCOUT )
      EQUIVALENCE (C(3338),LOSTAT)
      EQUIVALENCE (C(3339),NOSTAT)
      EQUIVALENCE (C(3440),NCRNOM)
       EQUIVALENCE (C(2008),PLOTNO)
       INTEGER PLOTNO
      INTEGER CPOINT
      LOSTAT = 0
      NOSTAT = 0
      NOSUB = 0
      NCMOD = 0
      NCOUT = 0
      NCRNOM = 0
      NOLIST = 0
      CPOINT=0
      NPLOT=0
      RETURN
      END
      SUBROUTINE    SUBL1
      COMMON C(4310)
      EQUIVALENCE (C(2461),NOSUB ), (C(2462),SUBNO )
      DIMENSION    SUBNO(99)
      DO 1 I = 1, NOSUB
      J = SUBNO(I)
      GO TO ( 1, 2, 3, 4, 5, 6, 7, 8, 9 ), J
    2 CALL INPT1
      GO TO 1
    3 CALL OUPT1
      GO TO 1
    4 CALL STGE1
      GO TO 1
    5 CALL CNTR1
      GO TO 1
    6 CALL RNCM1
      GO TO 1
    7 CALL AUXA1
      GO TO 1
    8 CALL AUXB1
      GO TO 1
    9 CALL AUXC1
    1 CONTINUE
      RETURN
```

```
      END
      SUBROUTINE   SUBL2
      COMMON C(4310)
      EQUIVALENCE (C(2461),NOSUB ), (C(2462),SUBNO )
      DIMENSION   SUBNO(99)
      DO 1 I = 1, NOSUB
      J = SUBNO(I)
      GO TO ( 1, 2, 3, 4, 5, 6, 7, 8, 9 ), J
    2 CALL INPT2
      GO TO 1
    3 CALL OUPT2
      GO TO 1
    4 CALL STGE2
      GO TO 1
    5 CALL CNTR2
      GO TO 1
    6 CALL RNCM2
      GO TO 1
    7 CALL AUXA2
      GO TO 1
    8 CALL AUXB2
      GO TO 1
    9 CALL AUXC2
    1 CONTINUE
      RETURN
       END
      SUBROUTINE   SUBL3
      COMMON C(4310)
      EQUIVALENCE (C(2461),NOSUB ), (C(2462),SUBNO )
      DIMENSION   SUBNO(99)
      DO 1 I = 1, NOSUB
      J = SUBNO(I)
      GO TO ( 1, 2, 3, 4, 5, 6, 7, 8, 9 ), J
    2 CALL INPT3
      GO TO 1
    3 CALL OUPT3
      GO TO 1
    4 CALL STGE3
      GO TO 1
    5 CALL CNTR3
      GO TO 1
    6 CALL RNCM3
      GO TO 1
    7 CALL AUXA3
      GO TO 1
    8 CALL AUXB3
      GO TO 1
    9 CALL AUXC3
    1 CONTINUE
      RETURN
       END
      SUBROUTINE STGE2
      COMMON C(4310)
      EQUIVALENCE (C(2011),KSTEP ), (C(2020),LCONV ), (C(2021),KCONV )
      KCONV = 0
      LCONV = 0
      KSTEP = 1
      RETURN
      END
      SUBROUTINE STGE3
      COMMON C(4310),GRAPH
      EQUIVALENCE (C(2000),T      ), (C(2001),TF     ), (C(2003),PCNT  )
```

```
      EQUIVALENCE (C(2010),STEP  ), (C(2011),KSTEP ), (C(2020),LCONV )
      EQUIVALENCE (C(2021),KCONV ), (C(2561),N     ), (C(2662),HMIN  )
      EQUIVALENCE (C(2663),HMAX  ), (C(2664),DER   ), (C(2765),EL    )
      EQUIVALENCE (C(2865),EU    ), (C(2965),VAR   )
      EQUIVALENCE (C(1973),KASE  ), (C(1974),NJ    ), (C(1975),NPT   )
      DIMENSION   DER(101)        , VAR(101)         , EL(100)
      DIMENSION EU(100),GRAPH(2,2)
      EXTERNAL AUXSUB
      CALL G4
      IF (ABS( T-TF) .LE. 0.01 ) GO TO 20
      IF ( (TP-T) .LT. 0.) GO TO 10
      IF (LCONV .EQ. 2) GO TO 20
      IF (LCONV .EQ. 1) GO TO 10
      IF(DER(1).LT.O.)DER(1)=-DER(1)*0.5
      RETURN
   10 IF(DER(1).GT.O.)DER(1)=-DER(1)*0.5
      KCONV = KCONV + 1
      IF(KCONV .GE. 10) GO TO 20
      RETURN
   20 PCNT = 1.0
C     DUMM = RECRM(LOC(X3),LOC(X4))
C     WRITE (6,30)  X3, X4
C  30 FORMAT (38H0 RESTART INITIALIZERS, X3 AND X4, ARE 2F11.0)
      IF(STEP .EQ.1.)GOTO 40
      PRECER = DER(1)
      DER(1) = 0.
      NJ=N-1
      NPT=0
      CALL AMRK(AUXSUB)
      DER(1) = PRECER
   40 CALL LUPT3
      KSTEP = 2
      RETURN
      END
      SUBROUTINE AMRK(AUXSUB)
C*SINGLE PRECISION VERSION* INDEPENDENT VARIABLE IN DOUBLE PRECISION
      COMMON C(4310),GRAPH
C     DOUBLE PRECISION DELT,TME
C     DOUBLE PRECISION NEWC(200),NEWP(200),OLD(200)
      DOUBLE PRECISION DELT,TME
      DOUBLE PRECISION NEWC(200),NEWP(200),OLD(200)
      DIMENSION T(1000)
      DIMENSION D(101)              ,EL(100)          ,EU(100)
      DIMENSION V(101),GRAPH(2,2)
C     DIMENSIONNEWP(200),OLD(200),NEWC(200)
      EQUIVALENCE (C(2662),HMIN  ), (C(2663),HMAX  ), (C(2664),D     )
      EQUIVALENCE (C(2765),EL    ), (C(2865),EU    ), (C(2965),V     )
      EQUIVALENCE (C(1971),RITE  ), (J1,N1)
      EQUIVALENCE (C(1973),KASE  ), (C(1974),NJ    ), (C(1975),NPT   )
      DATA KOUNT/0/
      DATA P1,P2,P3,P4/2.29166667,2.45833333,1.5416667,0.375/
      DATA C2,C3,C4/0.7916667,0.2083333,0.0416667/
      IF(KASE.GT.0)GO TO 20
      N1=NJ
      J2=J1+N1
      J3=J2+N1
      J4=J3+N1
      J5=J4+N1
      J6=J5+N1
      J7=J6+N1
      J8=J7+N1
      J9=J8+N1
```

```
      KASE=KASE+1
C*NPT.EQ.0  ADAMS-MOULTON INTEGRATION MODE
C*NPT.EQ.1  RUNGE-KUTTA INTEGRATION MODE
C*NPT.EQ.2  BEGINNING ADAMS-MOULTON WITH RUNGE-KUTTA START
   20 IF(NPT.EQ.1)GO TO 40
      IF(NPT.EQ.2)GO TO 30
      IF(SNGL(DELT).NE.(0.5*C(1)))GO TO 30
      IF(KOUNT.LT.3)GO TO 40
      GO TO 200
   30 KOUNT=0
      NPT=0
C*START RUNGE-KUTTA INTEGRATION
C*COMPUTE K1
   40 DO 50 I=1,N1
      NEWP(I)=V(I+1)
   50 CONTINUE
      TME=V(1)
      KOUNT=KOUNT+1
      DO 60 I=1,N1
   60 OLD(I)=C(1)*D(I+1)
C COMPUTE K2
      DELT=0.5*C(1)
      TME=TME+DELT
      V(1)=TME
      DO 70 I=1,N1
      IF(KOUNT.NE.2)GO TO 65
      K1=J9+I
      T(K1)=NEWP(I)
   65 T(I)=C(I+1)
      NEWP(I)=NEWP(I)+0.5*OLD(I)
   70 V(I+1)=NEWP(I)
      CALL AUXSUB
      DO 80 I=1,N1
   80 NEWC(I)=C(1)*D(I+1)
C COMPUTE K3
      DO 90 I=1,N1
      NEWP(I)=NEWP(I)+0.5*(NEWC(I)-OLD(I))
   90 V(I+1)=NEWP(I)
      CALL AUXSUB
      DO 100 I=1,N1
      K2=J7+I
  100 T(K2)=C(1)*D(I+1)
C COMPUTE K4
      TME=TME+DELT
      V(1)=TME
      DO 110 I=1,N1
      K2=J7+I
      NEWP(I)=NEWP(I)+T(K2)-0.5*NEWC(I)
  110 V(I+1)=NEWP(I)
      CALL AUXSUB
      DO 120 I=1,N1
      K3=J8+I
  120 T(K3)=C(1)*D(I+1)
C COMPUTE VALUE OF FUNCTION
      DO 130 I=1,N1
      K2=J7+I
      K3=J8+I
      NEWP(I)=NEWP(I)-T(K2)+0.16666667*
     X(OLD(I)+NEWC(I)+NEWC(I)+T(K2)+T(K2)+T(K3))
  130 V(I+1)=NEWP(I)
  140 CALL AUXSUB
      DO 150 I=1,N1
```

```
      K5=J1+I
      K0=J2+I
      K1=J3+I
      K2=J4+I
      K3=J5+I
      K4=J6+I
      T(K4)=T(K3)
      T(K3)=T(K2)
      T(K2)=T(K1)
      T(K1)=T(K0)
      T(K0)=T(K5)
      T(K5)=T(I)
  150 T(I)=C(I+1)
      RETURN
C ADAMS-MOULTON INTEGRATION
  200 KOUNT=KOUNT+1
      DELT=0.5*D(1)
      DO 210 I=1,N1
      K1=J2+I
      K2=J3+I
      K4=J1+I
C COMPUTE Y-PREDICTED
      OLD(I)=NEWP(I)
      NEWP(I)=OLD(I)+D(1)*(P1*T(I)-P2*T(K4)+P3*T(K1)-P4*T(K2))
  210 V(I+1)=NEWP(I)
      TME=TME+D(1)
      V(1)=TME
      CALL AUXSUB
      K5=0
      DO 250 I=1,N1
      K2=J2+I
      K4=J1+I
C COMPUTE Y-CORRECTED
      NEWC(I)=OLD(I)+D(1)*(P4*C(I+1)+C2*T(I)-C3*T(K4)+C4*T(K2))
      IF(HMIN.EQ.HMAX)GO TO 250
      TEMP=ABS(SNGL(NEWC(I)-NEWP(I)))
      IF(TEMP.LT.EU(I))GO TO 240
      IF(RITE.LE.0.0)GO TO 230
      SPTIME=SNGL(TME)
      WRITE(6,220)I,SPTIME,TEMP
  220 FORMAT(1H ,14HSTATE VARIABLE,I3,26H  EXCEEDED TOLERANCE ERROR,
     X7H  TIME=,F14.7,9H  TEMP=  ,1PE17.8)
  230 IF(ABS(SNGL(DELT)).GE.HMIN)GO TO 270
  240 IF(TEMP.LT.EL(I))K5=K5+1
  250 CONTINUE
      IF(K5.LT.N1)GO TO 290
      IF(ABS(D(1)+D(1)).GT.HMAX)GO TO 290
C SET-UP FOR DOUBLING STEP SIZE
      IF(KOUNT.LE.6)GO TO 290
      DO 260 I=1,N1
      K1=J1+I
      K2=J2+I
      K3=J3+I
      K5=J5+I
      T(I)=T(K1)
      T(K1)=T(K3)
  260 T(K2)=T(K5)
      D(1)=D(1)+D(1)
      KOUNT=4
      DELT=0.5*D(1)
      GO TO 290
C SET-UP FOR HALVING STEP SIZE
```

```
270 IF(KOUNT.LE.4)GO TO 310
    TME=TME-C(1)
    V(1)=TME
    D(1)=DELT
    DELT=0.5*D(1)
    DO 280 I=1,N1
    K1=J1+I
    K2=J2+I
    K3=J3+I
    NEWP(I)=OLD(I)
    V(I+1)=OLD(I)
    T(K3)=T(K2)+0.5*(T(K1)-T(K2))
    T(K2)=T(K1)
280 T(K1)=T(K1)+0.5*(T(I)-T(K1))
    KOUNT=4
    GO TO 200
C INTEGRATION IS FINISHED. SET UP DERIVATIVES AND EXIT.
290 DO 300 I=1,N1
    NEWP(I)=NEWC(I)
300 V(I+1)=NEWC(I)
    GO TO 140
C RETURN TO 3RD PRECEDING POINT AND RESTART RK
310 DO 320 I=1,N1
    K1=J9+I
    NEWP(I)=T(K1)
320 V(I+1)=T(K1)
    TME=TME-4.0*C(1)
    V(1)=TME
    D(1)=DELT
    CALL AUXSUB
    GO TO 30
    END
    SUBROUTINE AUXI
    COMMON C(4310)
    EQUIVALENCE (C(2361),NOMOD ), (C(2362),XMODNO), (C(2561),N      )
    DIMENSION  XMODNO(99)
    N = 1
    DO 1 I=1,NOMOD
    L =XMODNO(I)
    GO TO (1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23
   1      ,24,25,26,27,28,29,30,31,32,33,34,35,36,37),L
  2 CALL A1I
    GO TO 1
  3 CALL A2I
    GO TO 1
  4 CALL A3I
    GO TO 1
  5 CALL A4I
    GO TO 1
  6 CALL A5I
    GO TO 1
  7 CALL C1I
    GO TO 1
  8 CALL C2I
    GO TO 1
  9 CALL C3I
    GO TO 1
 10 CALL C4I
    GO TO 1
 11 CALL C5I
    GO TO 1
 12 CALL C6I
```

```
      GO TO 1
   13 CALL C7I
      GO TO 1
   14 CALL C8I
      GO TO 1
   15 CALL C9I
      GO TO 1
   16 CALL C10I
      GO TO 1
   17 CALL C1I
      GO TO 1
   18 CALL C2I
      GO TO 1
   19 CALL C3I
      GO TO 1
   20 CALL C4I
      GO TO 1
   21 CALL C5I
      GO TO 1
   22 CALL G1I
      GO TO 1
   23 CALL G2I
      GO TO 1
   24 CALL G3I
      GO TO 1
   25 CALL G4I
      GO TO 1
   26 CALL G5I
      GO TO 1
   27 CALL G6I
      GO TO 1
   28 CALL S1I
      GO TO 1
   29 CALL S2I
      GO TO 1
   30 CALL S3I
      GO TO 1
   31 CALL S4I
      GO TO 1
   32 CALL S5I
      GO TO 1
   33 CALL S6I
      GO TO 1
   34 CALL S7I
      GO TO 1
   35 CALL S8I
      GO TO 1
   36 CALL S9I
      GO TO 1
   37 CALL S10I
    1 CONTINUE
      RETURN
      END
      SUBROUTINE AUXSUB
      COMMON C(4310)
      EQUIVALENCE (C(2000),T      ), (C(2361),NOMOD ), (C(2362),XMCDNO)
      EQUIVALENCE (C(2561),N      ), (C(2562),IPL   ), (C(2664),DER   )
      EQUIVALENCE (C(2965),VAR    )
      EQUIVALENCE (C(2020),LCONV)
      DIMENSION   DER(101)         , VAR(101)         , IPL(100)
      DIMENSION   XMOCNO(99)
      DO 50 I = 2, N
```

```
      J = IPL(I-1)
   50 C(J+3) = VAR(I)
      T = VAR(1)
      DO 1 I=1,NOMOD
      IF(LCONV.EQ.2)RETURN
      L =IMCONC(I)
      GO TO (1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,
     123,24,25,26,27,28,29,30,31,32,33,34,35,36,37),L
    2 CALL A1
      GO TO 1
    3 CALL A2
      GO TO 1
    4 CALL A3
      GO TO 1
    5 CALL A4
      GO TO 1
    6 CALL A5
      GO TO 1
    7 CALL C1
      GO TO 1
    8 CALL C2
      GO TO 1
    9 CALL C3
      GO TO 1
   10 CALL C4
      GO TO 1
   11 CALL C5
      GO TO 1
   12 CALL C6
      GO TO 1
   13 CALL C7
      GO TO 1
   14 CALL C8
      GO TO 1
   15 CALL C9
      GO TO 1
   16 CALL C10
      GO TO 1
   17 CALL D1
      GO TO 1
   18 CALL D2
      GO TO 1
   19 CALL D3
      GO TO 1
   20 CALL D4
      GO TO 1
   21 CALL D5
      GO TO 1
   22 CALL G1
      GO TO 1
   23 CALL G2
      GO TO 1
   24 CALL G3
      GO TO 1
   25 CALL G4
      GO TO 1
   26 CALL G5
      GO TO 1
   27 CALL G6
      GO TO 1
   28 CALL S1
      GO TO 1
```

```
   29 CALL S2
      GO TO 1
   30 CALL S3
      GO TO 1
   31 CALL S4
      GO TO 1
   32 CALL S5
      GO TO 1
   33 CALL S6
      GO TO 1
   34 CALL S7
      GO TO 1
   35 CALL S8
      GO TO 1
   36 CALL S9
      GO TO 1
   37 CALL S10
    1 CONTINUE
      DO 60 I = 2, N
      J = IPL(I-1)
   60 DER(I) = C(J)
      RETURN
      END
      SUBROUTINE RESET
      COMMON C(4310)
      EQUIVALENCE(C(4310),KK),(C,K)
      EQUIVALENCE (C(3066),NOLIST), (C(3067),LISTNO), (C(3117),VALUE )
      EQUIVALENCE(C(2900),NORNOM),(C(3441),RNDMNO)
      EQUIVALENCE (C(2800),B)
      DIMENSION K(4310)
      DIMENSION    LISTNO(50)       , VALUE(50)
      DIMENSION B(100)
      DIMENSION RNDMNO(50)
      DO33I=1,3509
   33 C(I)=0.
      JC=3509+KK
      DO43I=3510,JC,2
   43 K(K(I+1))=K(I)
      K(2461)=K(4309)
      K(2361)=K(4308)
      K(3066)=K(4307)
      K(3167)=K(4306)
      K(3338)=K(4305)
      K(3339)=K(4204)
      K(1984)=K(4303)
      K(3440)=K(4302)
      K(2012)=C(2010)
      K(1975)=2
      RETURN
      END
C  DUMMY SUBROUTINE
      SUBROUTINE DUMMY
C     ENTRY A1
      ENTRY A1I
C     ENTRY A2
      ENTRY A2I
C     ENTRY A3
C     ENTRY A3I
      ENTRY A4
      ENTRY A4I
      ENTRY A5
      ENTRY A5I
```

```
C     ENTRY C1
C     ENTRY C1I
      ENTRY C2
      ENTRY C2I
      ENTRY C3
      ENTRY C3I
C     ENTRY C4
      ENTRY C4I
      ENTRY C5
      ENTRY C5I
      ENTRY C6
      ENTRY C6I
      ENTRY C7
      ENTRY C7I
      ENTRY C8
      ENTRY C8I
      ENTRY C9
      ENTRY C9I
      ENTRY C10
      ENTRY C10I
C     ENTRY D1
C     ENTRY D1I
C     ENTRY D2
C     ENTRY D2I
      ENTRY D3
      ENTRY D3I
      ENTRY D4
      ENTRY D4I
      ENTRY D5
      ENTRY D5I
      ENTRY G1
      ENTRY G1I
C     ENTRY G2
      ENTRY G2I
C     ENTRY G3
      ENTRY G3I
C     ENTRY G4
      ENTRY G4I
C     ENTRY G5
      ENTRY G5I
      ENTRY G6
      ENTRY G6I
C     ENTRY S1
C     ENTRY S1I
      ENTRY S2
      ENTRY S2I
      ENTRY S3
      ENTRY S3I
      ENTRY S4
      ENTRY S4I
      ENTRY S5
      ENTRY S5I
      ENTRY S6
      ENTRY S6I
      ENTRY S7
      ENTRY S7I
      ENTRY S8
      ENTRY S8I
      ENTRY S9
      ENTRY S9I
      ENTRY S10
      ENTRY S10I
```

```
      ENTRY AUXA1
      ENTRY AUXA2
      ENTRY AUXA3
      ENTRY AUXB1
      ENTRY AUXB2
      ENTRY AUXB3
      ENTRY AUXC1
      ENTRY AUXC2
      ENTRY AUXC3
      ENTRY CNTR1
      ENTRY CNTR2
      ENTRY CNTR3
      ENTRY IN  1
      ENTRY IN
      ENTRY IN
      ENTRY OUPT1
C     ENTRY OUPT2
C     ENTRY OUPT3
      ENTRY PROCES
C     ENTRY RESET
      ENTRY RNCH1
      ENTRY RNCH2
      ENTRY RNCH3
      ENTRY STGE1
C     ENTRY STGE2
C     ENTRY STGE3
C     ENTRY SUBL1
C     ENTRY SUBL2
      ENTRY KIKSET
      ENTRY COUNTW
      RETURN
      END
C**  FOR USE WITH COOIM2, FCN2, FCN3
C
      SUBROUTINE TERROR (XLABEL)
      COMMON C(4310)
      EQUIVALENCE (C(2020),LCONV)
      WRITE (6,10) XLABEL
 10   FORMAT ( 43H0          NO AERO POINTS SPECIFIED FOR ARG , 5X,
     C    7HTABLE  ,A6 )
      CALL EXIT
C
      ENTRY AERROR
      WRITE (6,20) XLABEL
   20 FORMAT  ( 43H0          OUT OF AERO TABLE ARGUMENT ARRAY , 5X,
     C    7HTABLE  ,A6 )
      DO 40 I=1202,1251,7
  40  WRITE(6,30) C(I),C(I+1),C(I+2),C(I+3),C(I+4),C(I+5),C(I+6)
   30 FORMAT(1H ,7E15.7)
      WRITE (6,30) C(2000),C(367),C(368),C(204),C(369),C(370),C(1117),
     C C(1118),C(1119),C(1120)
      LCONV=2
      RETURN
      END
      FUNCTION SIND (X)
      SIND= SIN (X/57.29578)
      RETURN
      END
      FUNCTION COSD (X)
      COSD= COS (X/57.29578)
      RETURN
      END
```

```
      FUNCTION ATAND (Y,X)
      IF(X.EQ.0..AND.Y.EQ.0.)GOTO1
      ATAND= 57.29578*ATAN2 (Y,X)
1     CONTINUE
      RETURN
      END
      SUBROUTINE TABLE (X,XI,YI,NX,XK,XLABEL,Y)
      DIMENSION XLABEL (2)
      Y  =    CCDIM2 (X,XI,YI,NX,XK,XLABEL)
      RETURN
      END
      SUBROUTINE TABL2(X,Y,XYI,ZI,NXY,XINTER,XLABEL,Z)
      DIMENSION XYI(2),NXY(2),ZI(2),XLABEL(2)
      Z=FCCDN2(X,Y,XYI(1),XYI(NXY(1)+1),ZI,NXY(1),NXY(2),NXY(1),XINTER,
      XLABEL)
      RETURN
      END
      SUBROUTINE . BL3(X,Y,Z,XYZI,WI,NXYZ,XINTER,XLABEL,W)
      DIMENSION XY I(4),NXYZ(4),WI(2),XLABEL(2)
      NZI= NXYZ(1) + NXYZ(2) + 1
      W=FCCDN3(X,Y,Z,XYZI(1),XYZI(NXYZ(1)+1),XYZI(NZI),WI,NXYZ(3),
      NXYZ(2),NXYZ(1),XINTER,XLABEL)
      RETURN
      END
      SUBROUTINE TIMEV(DELT)
      RETURN
      END
      SUBROUTINE WRITE(IA,P,N)
      RETURN
      END
      SUBROUTINE PLOT4(GRAPH,DPOINT,VLABLE,TIME,NPLOT4)
      RETURN
      END
      SUBROUTINE PLOT2(NPLOT2)
      RETURN
      END
      SUBROUTINE PLOTN(NCPLOT)
      RETURN
      END
C . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
C
C      SUBROUTINE CCDIM2
C
C         PURPOSE
C            TO FIT A SET OF POINTS WITH A CONTINUOUS FUNCTION THAT
C            SIMULATES A FRENCH CURVE TYPE CURVE FIT.
C
C         USAGE
C            Y  =  CODIM2 ( X , XI , YI , N , F , XLABEL )
C                OR
C            Y  =  CODIM1 ( X , XI , YI , N , F , XLABEL )
C
C         DESCRIPTION OF PARAMETERS
C            X          ARGUMENT - INDEPENDENT VARIABLE
C            XI         ARRAY OF INDEPENDENT VARIABLE , X
C            YI         ARRAY OF DEPENDENT VARIABLE , Y
C            N          NUMBER OF POINTS REPRESENTED BY XI AND YI ARRAYS
C            F          INTERPOLATION CONTROL
C                          LESS THAN ZERO - STRAIGHT LINE INTERPOLATION
C                          POSITIVE - END INTERVAL INTERPOLATION
C                              0.0   STRAIGHT LINE
C                              1.0   FULL PARABOLIC
C
```

```
C              XLABEL  HOLLERITH FIELD OF UP TO 6 CHARACTERS
C
C          REMARKS
C              EXTRAPOLATION IS DONE BY PASSING A STRAIGHT LINE THRU THE
C              TWO POINTS AT THE END INTERVAL.
C              THE ARRAY OF THE INDEPENDENT VARIABLE , XI , MAY BE IN
C              EITHER INCREASING OR DECREASING ORDER.
C
C . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .

      FUNCTION CODIM2 ( X , XI , YI , N , F , XLABEL )


      DIMENSION  XI(N) , YI(N) , P(2) , E(2) , IS(4,2) , XLABEL(2)
      LOGICAL  OUT
      DATA  IS / -1, 0, -2, -1,  0, 1, -1, 0 /

 100  OUT  =  .FALSE.
      N1   =  N
      XX   =  X
      J    =  1
      IF ( N1 - 2 )  150 , 1200 , 300
 150  CALL TERROR (XLABEL)
 200  CODIM2  =  YI(J)
      RETURN

 300  KPL  =  1
      KPU  =  2
      IF ( XI(1) - XI(2) )  400 , 150 , 600
 400  DO   500   J = 1 , N1
      IF ( XX - XI(J) )  900 , 200 , 500
 500  CONTINUE
      GO TO  800
 600  DO   700   J = 1 , N1
      IF ( XI(J) - XX )  900 , 200 , 700
 700  CONTINUE

 800  J  =  N1
      CALL AERROR (XLABEL)
      GO TO  1300

 900  OUT  =  F .LT. 0.0
      IF ( J - 2 )  1200 , 1000 , 1100
1000  KPL  =  2
      GO TO  1500
1100  IF ( J - N1 )  1500 , 1400 , 1300
1200  J  =  2
1300  OUT  =  .TRUE.
1400  KPU  =  1
1500  AL  =  ( XX - XI(J-1) ) / ( XI(J) - XI(J-1) )
      CODIM2  =  AL * YI(J) + ( 1.0 - AL ) * YI(J-1 )
      IF ( OUT )   RETURN

      DO    1800    KP = KPL , KPU
      P(KP)  =  0.0
      DO    1600    K = 1 , 3
      JO  =  J + KP + K - 4
      IF(JO)1900,1550,1900
1550  XC=0.
      YC=0.
      GOTO1950
1900  XC=XI(JO)
```

```
      YC=YT(J0)
 1950 J1=J+IS(K,KP)
      J2  =  J + IS(K+1,KP)
 1600 P(KP)  =  P(KP) + YII * ( XX - XI(J1) ) / ( XO - XI(J1) )
     *              * ( XX - XI(J2) ) / ( XO - XI(J2) )
      IF ( KPL .NE. KPU )   GO TO  1700
      J1  =  3 - KPL
      P(J1)  =  CODIM2 + F * ( P(KP) - CODIM2 )
      E(J1)  =  ABS ( P(J1) - CODIM2 )
 1700 E(KP)  =  ABS ( P(KP) - CODIM2 )
 1800 CONTINUE

      IF ( E(1) + E(2) .EQ. 0.0 )   RETURN
      CODIM2  =  ( ( E(1) * AL ) * P(2) + ( E(2) * ( 1.0 - AL ) )
     *              * P(1) ) / ( ( E(1) * AL ) + ( E(2) * ( 1.0 - AL ) ) )

      RETURN
      END
C     2-DIMENSIONAL INTERPOLATION SUBPROGRAM....FCOON2
C
C     CALLING SEQUENCE -
C     Z = FCOON2(X,Y,XI,YI,ZI,NXD,NY,NX,XK,XLABEL)
C
C          X = ARGUMENT - 1ST VARIABLE
C          Y = ARGUMENT - 2ND VARIABLE
C         XI = ARRAY OF 1ST VARIABLE
C         YI = ARRAY OF 2ND VARIABLE
C         ZI = ARRAY OF DEPENDENT VARIABLE
C        NXD = DIMENSIONED SIZE OF XI ARRAY
C         NY = NUMBER OF VALUES IN ARRAY YI
C         NX = NUMBER OF VALUES IN ARRAY XI
C         XK = END INTERVAL INTERPOLATION CONTROL CONSTANT
C     XLABEL = HOLLERITH FIELD OF UP TO 6 CHARACTERS
C
C         THIS ROUTINE DIFFERS FROM FCOOM2 IN THAT THE ZI ARRAY DOES NOT
C            HAVE TO BE PACKED - I.E.. IT DOES NOT HAVE TO OCCUPY CON-
C            SECUTIVE LOCATIONS IN CORE, AND IN THAT EITHER OR BOTH THE
C            XI AND YI ARRAYS MAY BE IN ASCENDING OR DESCENDING ORDER.
C
      FUNCTION FCOON2(X,Y,XI,YI,ZI,NXD,NY,NX,XK,XLABEL)
C
      DIMENSION XI(1), YI(1), ZI(NXD,1), T(4) , XLABEL(2)
C
      IF (NY.GT.4) GO TO 120
      N3=1
C         N3 IS THE INDEX NUMBER OF THE FIRST Y CURVE TO BE USED
      N4=NY
C         N4 IS THE COUNT OF THE NUMBER OF Y CURVES TO BE USED
      GO TO 200
C
 120  N4=4
      IF (YI(1)-YI(2)) 130,150,133
 130      DO 132 K=1,NY
          IF (Y-YI(K)) 150,150,132
 132      CONTINUE
      GO TO 140
C
C
 133      DO 134 K=1,NY
          IF (YI(K)-Y) 150,150,134
 134      CONTINUE
 140  N3=NY-3
```

```
      GO TO 200
C
 150  IF (K-3) 155,155,160
 155  N3=1
      GO TO 200
C
 160  IF (K-NY) 165,140,140
 165  N3=K-2
C
 200  L=N3
         DO 300 I=1,N4
         T(I)=CCOIM2(X,XI,ZI(1,L),NX,XK,XLABEL)
 300     L=L+1
C
      FCOON2=CCOIM2(Y,YI(N3),T,N4,XK,XLABEL)
      RETURN
      END
C     3-DIMENSIONAL INTERPOLATION SUBPROGRAM....FCOON3
C
C     CALLING SEQUENCE -
C
C     W = FCOON3(X,Y,Z,XI,YI,ZI,WI,NZ,NY,NX,XK,XLABEL)
C          X = ARGUMENT - 1ST VARIABLE
C          Y = ARGUMENT - 2ND VARIABLE
C          Z = ARGUMENT - 3RD VARIABLE
C         XI = ARRAY OF 1ST VARIABLE
C         YI = ARRAY OF 2ND VARIABLE
C         ZI = ARRAY OF 3RD VARIABLE
C         WI = ARRAY OF DEPENDENT VARIABLE
C         NZ = NUMBER OF POINTS IN ZI ARRAY
C         NY = NUMBER OF POINTS IN YI ARRAY
C         NX = NUMBER OF POINTS IN XI ARRAY
C         XK = END INTERVAL INTERPOLATION CONTROL CONSTANT (0.0 TO 1.0)
C     XLABEL = HOLLERITH FIELD OF UP TO 6 CHARACTERS
C
C         FCOON3 DIFFERS FROM FCOOM3 IN THAT THE WI ARRAY DOES NOT NEED
C           TO BE PACKED, I.E., WI NEED NOT OCCUPY CONSECUTIVE LOCATIONS
C           IN CORE, AND ANY OR ALL ARRAYS MAY BE IN EITHER ASCENDING OR
C           DESCENDING ORDER.
C
      FUNCTION FCOON3(X,Y,Z,XI,YI,ZI,WI,NZ,NY,NX,XK,XLABEL)
C
      DIMENSION XI(1), YI(1), ZI(1), W(1,1,1), T(4) , XLABEL(2)
C
      IF (NZ.GT.4) GO TO 120
      N4=1
      N5=NZ
      GO TO 200
C
 120  N5=4
      IF (ZI(1)-ZI(2)) 130,150,133
 130     DO 132 K=1,NZ
         IF (Z-ZI(K)) 150,150,132
 132     CONTINUE
      GO TO 140
C
 133     DO 134 K=1,NZ
         IF (ZI(K)-Z) 150,150,134
 134     CONTINUE
 140  N4=NZ-3
      GO TO 200
C
```

```
 150   IF (K.GT.3) GO TO 160
       N4=1
       GC TO 200
C
 160   IF (K.GE.NZ) GO TO 140
       N4=N-2
C
 200   L=N4
          CC 300 I=1,NS
       F(I)= FCOCN2 (X,Y,XI,YI,WI(1,1,L),NX,NY,NX,XK,XLABEL)
 300      L=L+1
C
       FCCCN3=CCOIN2(Z,ZI(N4),T,NZ,XK,XLABEL)
       RETURN
       END

 1      CUPT 2,3             3
 1      BTGB 2,3             4
 2      G2-T                23
 2 G3-H                     24
 2 G5-H                     26
 2      S1-T                28
 2      C1-T                 7
 2      C4-T                10
 2      A1-T                 2
 2      A3-T                 4
 2 A2-H                      3
 2 C1-H                     17
 2  C-10 H                  16
 2 C2-H                     18
 3  TF                    2001      7.0
 3  UCC                   2013      7.
 3  HMIN                  2652      .CC75
 3  HMAX                  2663      .CO75
 3  CERI                  2664      .CO25
 3 CPIKR                   443      1.
 3 CKDT                    445      .05
 3 GSW                     450      1.
 3  RBKLCK                 453      1CO.
 3 CKSKR                   456      2.
 3 VZE                    1611      110.
 3 VXE                    1603      440.
 3  BDIVE                 1666     -20.          1.
 3  RSLANT                1667      9CCO.
 3  VMACH                  204      .82789       1.
 3  THETA-G                427      .1
 3  BALPHA                 367      2.3
 3  CBIAS                  852      1.
 3  HLIMC                  850      15.
 3  HLIMC                  851      15.
 3     GY                  856      6.0
 3 GZ                      855     3.0
 3     TAUY                864      6.0
 3     TAUZ                863      6.0
 3 TAUL                    877      .5
 3  BCMAX                 1121      20.
 3 RFAREA                 1306      .267
 3 RFLGTH                 1307      .584
 3 RFXCG                  1313      2.75
 3 RFZCG                  1315      -.00833
 3 RLCGC                  1421      2.54
 3 FMXYO                  1420      15.4
```

```
3 ROCGP                  1418        -.1165
3 ROCGC                  1417        -.3165
3 DWP                    1416        19.4
3 DWT                    1415        135.
3 CISP                   1414        195.8
3 VAIRSPD                 207        880.
3  CKSK2                  888        6.
3 GNALGA                 1403        1.
3 RAIL                   1317        7.5
3 RLUG                   1316        2.57
3  CEFOCS                 468        3.
3  CETRAD                 469        7.5
3  CKACT                 1163        1.
3  TDY2                   866        1.
3   TCLY2                 866        9.0
3  CKSK1                  472        1.0
3  VLAZRP                 473        1.00
3  CKSK2                  888        .6
3  CER1                  2664        .C075
3  FMIXC                 1419        .234
3    GZ                   855        6.0
3 AGRAV                  1627        32.174
3 CRAD                   1751        57.295778       -0.
3 STEP                   2010         2.             -0.
3 BCRE                  00C01        0.10            1.0
3 RADIUS                0C003        0.250           1.0
3 WAND                  0C002         0.20                 1.0
3 AIFAC                    11         0.0
3 AILL                      7         1.0
3 SPLTMC                    8         1.0            1.0
3  CPP                   2015        1.C
4         P              1739
4         RCLL FLAP      1232
4         PHI MISSILE     352
4         VX EARTH       1603
4         X MSL          1615
4         C              1743
4         PITCH FLAP     1233
4         THETA MISSLE    350
4         VY EARTH       1607
4         Y MSL          1619
4         R              1747
4         YAW FLAP       1234
4         PSI MISSILE     351
4         VZ EARTH       1611
4         Z MSL          1623
4         ALPHA DEG       367
4         BETA CEG        368
4         ALPHA PRIME     369
4         PHI PRIME       370
4         AZEA           1626
4         MACH            204
4         AIR SPEED       207
4         PDYNMC          203
4         FTHRST         1410
4         THETA LCS-V     363
4         EPS Z           435
4         THETA GIMBAL    427
4         C CCM           869
4         PSI LCS-V       364
4         EPS Y           436
4         PSI GIMBAL      431
```

```
4       R CCM         870
4       CELT P COM    875
4       GAMMAW        357
4       GAMMAV        358
4       AREA          1625
4       EZ            403
4       EY            407
4    RY               405
4    RZ               404
6                      -0      -C.            -0.
3    RSLANT           1667     5C00.0
3   CPP               2015     .1
3     BDIVE           1666     -30.
6                      -0      -0.            -0.
3 GZ                  855     5.0
3 GY                  856     5.0
6                      0       0.
7
7
```

TIME	P	ROLL FLAP	PHI MISSILE	VX EARTH	X MSL
	Q	PITCH FLAP	THETA MISSILE	VY EARTH	Y MSL
	R	YAW FLAP	PSI MISSILE	VZ EARTH	Z MSL
	ALPHA DEG	BETA DEG	ALPHA PRIME	PHI PRIME	AZEA
	MACH	AIR SPEED	PDYNMO	FTHRST	THETA LOS-V
	EPS Z	THETA GIMBAL	G COM	PSI LOS-V	EPS Y
	PSI GIMBAL	R COM	DELT P COM	GAMMAH	GAMMAV
	AYDA	EZ	EY		
.0075000	0.	-5.5511151E-17	0.	6.0000117E+02	-1.0332335E+03
	0.	-3.7100000E-02	-2.7700000E+01	0.	0.
	0.	-3.3306691E-16	0.	4.6184302E+02	-5.3053763E+02
	2.2962775E+00	0.	2.2962775E+00	0.	-2.8486655E+01
	9.2850007E-01	9.2374574E+02	9.9445326E+02	7.9085000E+02	-1.7273522E-03
	-4.8682251E-06	-2.3000000E+00	1.0000000E+00	0.	-1.5257373E-15
	1.9257373E-15	7.6493975E-15	0.	0.	-2.9558277E+01
	0.	-2.9234160E-12	2.4411607E-13	0.	0.
.1050000	3.0877213E-02	8.4326262E-02	-3.7012325E-01	6.7669356E+02	-9.5193056E+02
	-2.0890733E+01	-1.2103549E+00	-2.3180363E+01	-6.9345375E-01	-5.9187567E+02
	1.2050093E+01	6.7955105E-01	3.3740641E-01	5.0496441E+02	-5.4570576E+02
	-3.7000203E-01	-5.3901273E-02	3.7221321E-01	-1.7478399E+02	1.5554135E+01
	3.3965243E-01	1.0473436E+03	1.2815056E+03	3.5574230E+03	-6.3307292E-02
	-4.4684736E-03	-8.1793895E-01	9.9040495E-01	-4.2382390E-02	-2.8497262E-03
	-3.1697732E-01	1.6071336E-03	6.4054061E-02	-4.5320724E-02	-2.3941475E+01
	2.4022153E+00	-5.0930355E-02	3.4631390E-02	0.	0.
.2025000	3.6456734E+01	4.3914636E-02	-4.7510322E+01	9.5065392E+02	-5.6279066E+02
	3.5450003E+00	-3.2490069E-01	-2.9701793E+01	2.4411772E+00	1.3593592E-02
	-7.6173473E+00	5.2993034E-02	1.3454000E+00	5.5305305E+02	-4.5030616E+02
	-6.2952323E-01	-2.8477347E-01	6.6635716E-01	-1.5522346E+02	2.3032575E+01
	1.0183305E+00	1.1363568E+03	1.5037535E+03	3.7974305E+03	2.2552316E-01
	7.4033543E-03	-2.2049076E-01	7.4031336E-01	1.2769926E-01	-4.2912193E-03
	-1.1643713E+00	1.7665410E-01	4.3335692E-02	1.4709314E-01	-3.0211154E+01
	1.2149337E+01	7.1252616E-02	-4.3077564E-02	0.	0.
.3000000	3.3026097E-01	-9.2514131E-03	-5.7028050E-02	1.0243332E+03	-7.6651576E+02
	-2.3613903E+00	-9.7192157E-01	-2.0555600E+01	2.9503309E+00	3.1122737E+01
	1.4401255E+00	1.0103407E-01	5.5931111E-01	5.9362400E+02	-4.4190441E+02
	3.2780035E-01	3.5697708E-01	-4.3010590E-01	4.7593394E+01	-1.4145636E+01
	1.0950535E+00	1.2222612E+03	1.7435199E+03	3.0250066E+03	3.0347305E-01
	3.3702234E-02	-1.0320014E+00	1.0011500E+00	1.6297779E-01	-1.9103395E-02
	-4.0367450E-01	-1.3133904E-02	-9.2529557E-03	1.6544305E-01	-3.0273346E+01
	-1.6444364E+01	7.7292010E-02	-4.6077509E-02	0.	0.

.4650000	1.5352200E-01	5.7243103E-02	-3.6730061E-01	1.1033255E+03	-6.5473367E+02
	-1.1013034E+00	-6.6513420E-01	-2.4757900E+01	4.2547137E+00	6.4763777E-01
	-3.1125537E-01	3.4724254E+01	1.2265987E+00	6.4061459E+02	-3.7659402E+02
	-3.1399623E-01	-2.2515281E-01	3.0037092E-01	-1.4435707E+02	1.9772693E+01
	1.1780204E+00	1.3133558E+03	2.0333211E+03	3.3305335E+03	6.8332718E-01
	5.1701393E-02	-1.3691930E-01	3.2121122E+00	2.3944776E-01	-2.3390473E-02
	-1.3692073E+00	-1.3691308E+00	5.7223165E-02	2.2269370E-01	-3.0369031E+01
	1.3155051E+01	5.1260152E-01	-2.7573280E-01	0.	0.
.5025000	-1.1191463E-01	-3.7352277E-04	-1.1310446E-01	1.1776406E+03	-5.4357491E+02
	-2.4170277E-02	-9.7340979E-01	-2.9486593E-01	9.9373307E+00	1.1545535E+00
	-1.1570035E+00	3.1021607E-01	7.3389621E-01	6.5242766E+02	-3.1126172E+02
	1.1837453E+01	2.0592931E+01	2.3792777E-01	6.6108751E+01	-3.5078397E+00
	1.2557804E+00	1.4015511E+03	2.3107720E+03	3.0522131E+03	6.6900402E-01
	1.0894021E-01	-4.1910003E-01	5.0135664E+00	3.3725831E-01	-5.3635213E-02
	-6.3050713E-01	-2.6933640E+00	-3.5135546E-04	2.6556772E-01	-3.0467117E+01
	-1.1857649E+01	1.1502165E+00	-6.1563360E-01	0.	0.

TIME	P	ROLL FLAP	PHI MISSILE	VX EARTH	X MSL
	Q	PITCH FLAP	THETA MISSILE	VY EARTH	Y MSL
	R	YAW FLAP	PSI MISSILE	VZ EARTH	Z MSL
	ALPHA DEG	BETA DEG	ALPHA PRIME	PHI PRIME	AZBA
	MACH	AIR SPEED	PDYNMC	FTHRST	THETA LOS-V
	EPS Z	THETA GIMBAL	Q COM	PSI LOS-V	EPS Y
	PSI GIMBAL	R COM	DELT P COM	GAMMAH	GAMMAV
	AYDA	EZ	EY		
.6000000	-1.4722082E-01	4.2630141E-02	-2.7495305E-01	1.2496533E+03	-4.2929787E+02
	-2.5935333E-02	-1.4099434E+00	-2.9713751E+01	6.3715013E+00	1.7125230E+00
	1.3546301E+00	6.1028955E-01	1.0444306E+00	7.3724361E+02	-2.4159352E+02
	5.1153075E-03	-8.1555952E-02	6.1734790E-02	-6.5712794E+01	3.6996240E+00
	1.3318403E+00	1.4603129E+03	-2.6044471E+03	3.2414179E+03	-9.3860120E-01
	2.0201620E-01	-8.4863762E-02	1.0360909E+01	4.5042360E-01	-3.9483540E-02
	-1.0167403E+00	-5.3783977E+00	-4.3051035E-02	2.9213198E-01	-3.0539052E+01
	7.4450353E+00	1.6502165E+00	-6.1553300E-01	0.	0.
.7050000	1.7260423E+00	-6.6143026E-03	-3.1514138E-02	1.3265327E+03	-2.9600313E+02
	3.2655340E+00	-1.5003653E+00	-2.9445435E+01	7.5515310E+00	2.4553843E+00
	-1.2625330E+00	8.6562344E-01	7.5650149E-01	7.5427404E+02	-1.6150475E+02
	3.7532173E-01	1.6427390E-01	4.1244787E-01	2.3472036E+01	-1.1502617E+01
	1.4121163E+00	1.5703767E+03	2.3599058E+03	3.6219317E+03	1.4534355E+00
	-4.9214362E-01	-1.5093150E-01	-1.3258111E+01	7.0676957E-01	-2.0795055E-01
	-3.7630313E-01	-9.6310500E+00	-6.6110617E-03	3.2015063E-01	-3.0590956E+01
	-1.0264933E+01	4.3878275E+00	-1.1071331E+00	0.	0.
.8029000	-2.3900363E+00	-3.2435770E-02	1.3111726E-01	1.3976176E+03	-1.5715603E+02
	1.1707474E+01	-2.9303573E+00	-2.8830735E+01	7.7567357E+00	3.2090652E+00
	-1.7561355E+00	1.3199112E+00	7.3337410E-01	8.2460275E+02	-8.3214744E+01
	9.7532734E-01	1.5375955E-01	9.6737290E-01	6.9570330E+00	-4.1574858E+01
	1.4847515E+00	1.6538859E+03	3.2033195E+03	3.7593319E+03	2.6610531E+00
	1.2096427E+00	-2.7057622E-01	3.5332956E+01	1.2826066E+00	-7.0032437E-01
	-3.7621335E+01	-1.3645522E+01	-3.2428210E-02	3.3756377E-02	-3.0542917E+01
	-6.5615000E+00	9.4050710E+00	-5.3672706E+00	0.	0.

.9000000	7.5790330E+00	-1.5035594E-01	7.7934962E-01	1.4703534E+03	-1.7413955E+01
	-3.0207724E+01	-4.6003109E+00	-2.0573355E+01	6.5695309E+00	-3.9463425E+00
	-7.7070307E+00	2.3604765E+00	2.7742113E-01	6.5721249E+02	-1.1221247E+00
	-2.2075695E+00	-5.2143181E+01	-2.7183560E+00	3.3052736E+01	1.4759213E+02
	1.5555135E+00	1.7373066E+03	3.5030732E+03	3.7751716E+03	2.7301761E+01
	-2.8060035E+01	-1.3177480E+00	8.0503175E+01	1.2812374E+01	-5.4209671E-20
	-1.3775157E+00	-3.5043441E+01	-1.5035363E-01	2.7150476E-01	-3.0241616E+01
	-3.0092765E+01	-9.4058712E+00	-5.3072706E+00	0.	0.

BREAKLOCK AT RANGE EQUAL (J) 1.7691E+01

BREAKLOCK HAS OCCURRED, TIME= .9037, RANGE= 12.7196

MISS DISTANCE= 3.7657663E+00

TIME FINAL= 9.0314642E-01

XM EARTH= -3.5341509E+00 YM EARTH= 4.0110152E+00 ZM EARTH= 6.7264846E+00

Y FLIPATH= 5.4444017E+00 Z FLIPATH= 6.7345301E+00

(The reverse of this page is blank)

APPENDIX I

COORDINATE TRANSFORMATION FROM BODY TO GIMBAL AXIS SYSTEM

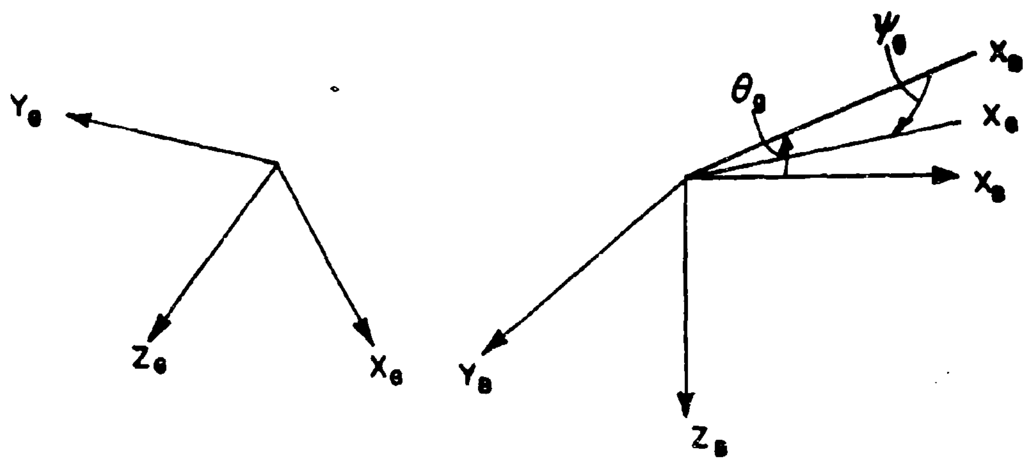


Figure I-1. Angles Between Gimbal and Body Axes

Transformation for Gimbal Pitch Angle (θ_g)

$$\begin{bmatrix} X_B' \\ X_B' \\ Z_B' \end{bmatrix} = \begin{bmatrix} \cos\theta_g & 0 & -\sin\theta_g \\ 0 & 1 & 0 \\ \sin\theta_g & 0 & \cos\theta_g \end{bmatrix} \begin{bmatrix} X_B \\ Y_B \\ Z_B \end{bmatrix}$$

Transformation for Gimbal Yaw Angle (ψ_g)

$$\begin{bmatrix} X_G \\ Y_G \\ Z_G \end{bmatrix} = \begin{bmatrix} \cos\psi_g & \sin\psi_g & 0 \\ -\sin\psi_g & \cos\psi_g & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_B' \\ Y_B' \\ Z_B' \end{bmatrix}$$

Transformation from Body Axis to Gimbal Axes

$$\begin{bmatrix} X_G \\ Y_G \\ Z_G \end{bmatrix} = \begin{bmatrix} \cos\psi_g & \sin\psi_g & 0 \\ -\sin\psi_g & \cos\psi_g & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\theta_g & 0 & -\sin\theta_g \\ 0 & 1 & 0 \\ \sin\theta_g & 0 & \cos\theta_g \end{bmatrix} \begin{bmatrix} X_B \\ Y_B \\ Z_B \end{bmatrix}$$

Transformation for both Gimbal Pitch angle and Gimbal Yaw angle for Range Determination is given as:

$$\begin{bmatrix} RXG \\ RYG \\ RZG \end{bmatrix} = \begin{bmatrix} \cos\psi\cos\theta & \sin\psi & -\sin\theta\cos\psi \\ -\sin\psi\cos\theta & \cos\psi & -\sin\psi\sin\theta \\ \sin\theta & 0 & \cos\theta \end{bmatrix} \begin{bmatrix} RXBA \\ RYBA \\ RZBA \end{bmatrix}$$

APPENDIX II

HIGH FREQUENCY ACTUATOR PROGRAM LISTING

```
C** INITIALIZATION MODULE FOR  TIGER   SIMPLIFIED ACTUATOR
C******HIGH FREQUENCY MODEL******
C
      SUBROUTINE C4I
      COMMON C(4310)
      DIMENSION IPL(100)
      DIMENSION BDELT(4)
      EQUIVALENCE (C(1116),NMDELT)
      EQUIVALENCE(C(1160),DELTPB),(C(1161),DELTQB)
      EQUIVALENCE(C(1162),DELTRB)
      EQUIVALENCE (C(2561),N      )
      EQUIVALENCE (C(2562),IPL    )
C
      NMDELT=N
      BDELT(1) = C(1103)
      BDELT(2) = C(1107)
      BDELT(3) = C(1111)
      BDELT(4) = C(1115)
      BDELT(1)  =  BDELT(1) - DELTPB + DELTQB - DELTRB
      BDELT(2)  =  BDELT(2) - DELTPB + DELTQB + DELTRB
      BDELT(3)  =  BDELT(3) + DELTPB + DELTQB - DELTRB
      BDELT(4)  =  BDELT(4) + DELTPB + DELTQB + DELTRB
      C(1103) = BDELT(1)
      C(1107) = BDELT(2)
      C(1111) = BDELT(3)
      C(1115) = BDELT(4)
      IPL(N) = 1100
      IPL(N+1) = 1104
      IPL(N+2) = 1109
      IPL(N+3) = 1112
      IPL(N+4) = 1124
      IPL(N+5) = 1128
      IPL(N+6) = 1132
      IPL(N+7) = 1136
      IPL(N+8)=1140
      IPL(N+9)=1144
      IPL(N+10)=1148
      IPL(N+11)=1152
      N=N+12
      CALL KIKSET
      RETURN
      END
C**    TIGER SIMPLIFIED ACTUATOR MODEL
C******HIGH FREQUENCY MODEL******
C
      SUBROUTINE C4
C
      COMMON C(4310)
      DIMENSION BDELTC(4),BDELT(4),BDELTC(4),VAR(101)
      DIMENSION BDELTP(4),BDLTCD(4)
      DIMENSION BDELTC(4)
      DIMENSION FMH(4),BDLTC3(4)
C
C**INPUT DATA
      EQUIVALENCE(C(1163),CKACT)
      EQUIVALENCE(C(1309),FMH1),(C(1310),FMH2),(C(1311),FMH3),(C(1312),
     *FMH4)
      EQUIVALENCE (C(1121),BDMAX )
      EQUIVALENCE (C(1122),CACTO )
      EQUIVALENCE(C(1160),DELTPB),(C(1161),DELTQB)
      EQUIVALENCE(C(1162),DELTRB)
C       BDELT ARE MEASURED IN RADIANS
```

```
C
C**INPUTS FROM OTHER MODULES
      EQUIVALENCE (C(1116),NHDELT)
      EQUIVALENCE (C( 857),DELTC)
      EQUIVALENCE (C(1117),BSURF1)
      EQUIVALENCE (C(1118),BSURF2)
      EQUIVALENCE (C(1119),BSURF3)
      EQUIVALENCE (C(1120),BSURF4)
C
C**STATE VARIABLE OUTPUTS
      BDELTP(1) = C(1127)
      BDELTP(2) = C(1131)
      BDELTP(3) = C(1135)
      BDELTP(4) = C(1139)
      BDELT(1)=C(1103)
      BDELT(2)=C(1107)
      BDELT(3)=C(1111)
      BDELT(4)=C(1115)
      PT=6.28318/360.
      FMH(1)=FMH1
      FMH(2)=FMH2
      FMH(3)=FMH3
      FMH(4)=FMH4
C
      A1=1./16.3/180./180.
      A4=1.
      A3=.6/180.+1./16.3
      A2=1./160./180.+.6/16.3/180.
      B1=CKACT
      BT=ELMAX*PT
C**INPUTS FROM MAIN PROGRAM
      EQUIVALENCE (C(2965),VAR    )
      C(1100)=C(1127)
      C(1112)=C(1139)
      C(1108)=C(1135)
      C(1104)=C(1131)
      C(1136)=C(1155)
      C(1132)=C(1151)
      C(1128)=C(1147)
      C(1124)=C(1143)
C
C**FLAP DEFLECTION BIAS
      BDELTC(1) = BDELTC(1) - DELTPB + DELTQB - DELTRB
      BDELTC(2) = BDELTC(2) - DELTPB + DELTQB + DELTRB
      BDELTC(3) = BDELTC(3) + DELTPB + DELTQB - DELTRB
      BDELTC(4) = BDELTC(4) + DELTPB + DELTQB + DELTRB
C
C**ACTUATOR DYNAMICS
      DO 30  I=1,4
      BDELTC(I)=BDELTC(I)*PT
      K=(I-1)*4+1
      BDLTCC(I)=C(1142+K)
      BDELTD(I) = BDELTP(I)
      BDLTD3(I)=-A2/A1*BDLTDD(I)-A3/A1*BDELTD(I)-A4/A1*BDELT(I)
     *+BDELTC(I)/A1*B1*FMH(I)/A1
C
C**SURFACE POSITION LIMITER
      IF (ABS(BDELT(I)) .LT. BT ) GO TO 30
      BDELT(I) = SIGN(BT ,BDELT(I))
      J = NHDELT + I
      VAR(J) = BDELT(I)
      IF (SIGN(1., BDELTP(I)) .NE. SIGN(1., BDELT(I))) GO TO 30
```

```
      BCELTC(I) = 0.
      BCELTP(I) = 0.
      J = J + 4
      VAR(J) = BCELTP(I)
   30 CONTINUE
C
      BSURF1=BCELT (1)/PT
      BSURF2=BCELT (2)/PT
      BSURF3=BCELT (3)/PT
      BSURF4=BCELT (4)/PT
      C(1103) = BCELT(1)
      C(1107) = BCELT(2)
      C(1111) = BCELT(3)
      C(1115) = BCELT(4)
      C(1140)=BCLTC3(1)
      C(1144)=BCLTC3(2)
      C(1143)=BCLTC3(3)
      C(1152)=BCLTC3(4)
C
C**OUTPUT DERIVATIVES OF STATE VARIABLES TO INTEGRATION
      C(1100) = BCELTC(1)
      C(1104) = BCELTC(2)
      C(1108) = BCELTC(3)
      C(1112) = BCELTC(4)
      C(1127)=C(1100)
      C(1139)=C(1112)
      C(1135)=C(1108)
      C(1131)=C(1104)
      RETURN
      END
```

APPENDIX III

HIGH FREQUENCY AUTOPILOT PROGRAM LISTING

```
C**    TIGER AUTOPILOT INITIALIZATION MODULE
C*******HIGH FREQUENCY MODEL*******
      SUBROUTINE C1I
      COMMON C(4310)
      DIMENSION IPL(100)
      EQUIVALENCE (C( 839),EZS    )
      EQUIVALENCE (C( 843),EYS    )
      EQUIVALENCE (C( 883),EZSS   )
      EQUIVALENCE (C( 807),EYSS   )
      EQUIVALENCE (C( 464),CCAMVS)
      EQUIVALENCE (C( 465),CCAMHS)
      EQUIVALENCE (C(3504),CPTN4 )
      EQUIVALENCE (C(0848),NFSUM )
      EQUIVALENCE (C(0956),GY     )
      EQUIVALENCE (C(2561),N      )
      EQUIVALENCE (C(2562),IPL    )
C
      NFSUM = N
      IPL(N) = 800
      IPL(N+1) = 804
      IPL(N+2) = 808
      IPL(N+3) = 812
      IPL(N+4) = 816
      IPL(N+5) = 820
      IPL(N+6) = 824
      IPL(N+7) = 828
      IPL(N+8) = 832
      IPL(N+9) = 836
      IPL(N+10) = 840
      IPL(N+11) = 880
      IPL(N+12) = 884
      IPL(N+14)=916
      IPL(N+13)=912
      IPL(N+15)=908
      IPL(N+16)=890
      IPL(N+17)=894
      IPL(N+18)=900
      IPL(N+19)=904
      N=N+20
   21 EZS = CCAMVS
      EYS = CCAMHS
   22 EZSS = EZS
      EYSS = EYS
      C( 803) = 0.
      C( 807) = 0.
      C( 811) = 0.
      C( 815) = 0.
      C( 819) = 0.
      C( 823) = 0.
      C( 827) = 0.
      C( 831) = 0.
      C( 839) = 0.
      RETURN
      END
C**    TIGER AUTOPILOT MODULE
C******HIGH FREQUENCY MODEL******
      SUBROUTINE C1
      COMMON C(4310)
      DIMENSION BDELTC(4),VAR(101)
C
C**INPUT DATA
      EQUIVALENCE (C(0850),HLIMO )
```

```
      EQUIVALENCE (C(0851),HLIME )
      EQUIVALENCE (C( 852),CBIAS )
      EQUIVALENCE(C(1171),DUMA4)
      EQUIVALENCE (C( 853),RBIAS )
      EQUIVALENCE(C(919),URS),(C(916),URSD),(C(908),DDELPO)
      EQUIVALENCE(C(915),UOS),(C(912),UOSD),(C(911),DDELFC)
      EQUIVALENCE(C(893),EUCCR),(C(897),EUDUC),(C(903),EVNCR)
      EQUIVALENCE(C(907),EVNCL),(C(890),EIDCR),(C(894),EIDDC)
      EQUIVALENCE(C(900),EVNCR),(C(904),EVNUL)
      EQUIVALENCE (C(0858),CZ     )
C     C( 857) THRU C( 860) ARE USED BY ECNTRL(I)
      EQUIVALENCE (C(0863),TAUZ   )
      EQUIVALENCE (C(0864),TAUY   )
      EQUIVALENCE (C( 865),TCY1   )
      EQUIVALENCE (C( 866),TCY2   )
      EQUIVALENCE (C( 877),TAUL   )
C
C**INPUTS FROM OTHER MODULES
      EQUIVALENCE (C(0352),BPHI   )
      EQUIVALENCE (C(0355),BPHID  )
      EQUIVALENCE (C(0403),EZ     )
      EQUIVALENCE (C(0407),EY     )
      EQUIVALENCE (C(0048),NFSUM  )
      EQUIVALENCE (C(1739),WP     )
      EQUIVALENCE (C(1740),WCD    )
      EQUIVALENCE (C(1743),WC     )
      EQUIVALENCE (C(1744),WRD    )
      EQUIVALENCE (C(1747),WR     )
      EQUIVALENCE (C(1751),CRAD   )
C
C**INPUTS FROM MAIN PROGRAM
      EQUIVALENCE (C(2000),T      )
      EQUIVALENCE (C(2965),VAR    )
      EQUIVALENCE (C(2664),DER    )
C
C** STATE VARIABLE OUTPUTS
      EQUIVALENCE (C( 800),BPHISD)
      EQUIVALENCE (C( 803),BPHIS )
      EQUIVALENCE (C( 804),WCSCD )
      EQUIVALENCE (C( 807),WCSP  )
      EQUIVALENCE (C( 808),WCSC  )
      EQUIVALENCE (C( 811),WCS   )
      EQUIVALENCE (C( 812),WRSCD )
      EQUIVALENCE (C( 815),WRSP  )
      EQUIVALENCE (C( 816),WRSC  )
      EQUIVALENCE (C( 819),WRS   )
      EQUIVALENCE (C( 820),ESUMDC)
      EQUIVALENCE (C( 823),ESUMD )
      EQUIVALENCE (C( 824),ESUMEC)
      EQUIVALENCE (C( 827),ESUME )
      EQUIVALENCE (C( 828),EZSCD )
      EQUIVALENCE (C( 831),EZSP  )
      EQUIVALENCE (C( 832),EZSC  )
      EQUIVALENCE (C( 835),EZS   )
      EQUIVALENCE (C( 836),EYSCD )
      EQUIVALENCE (C( 839),EYSP  )
      EQUIVALENCE (C( 840),EYSC  )
      EQUIVALENCE (C( 843),EYS   )
      EQUIVALENCE (C( 8[illegible]),EZSSD )
      EQUIVALENCE (C( 8[illegible]),EZSS  )
      EQUIVALENCE (C( 884),EYSSD )
      EQUIVALENCE (C( 887),EYSS  )
```

```
C
C**OUTPUTS
      EQUIVALENCE (C( 857),BDELTC)
C
C**OTHER OUTPUTS
      EQUIVALENCE (C(0876),EZRR   )
      EQUIVALENCE (C(0868),EYRR   )
      EQUIVALENCE (C( 859),WCC    )
      EQUIVALENCE (C( 870),WRC    )
C
C**GUIDANCE SIGNAL SHAPING
      ELCCR=ECCCC
      EVNCR=EVACD
      EZSC = EZSP
      EYSC = EYSP
      EZSCC = TAUZ*(TAUZ*(GZ*EZ - EZS) - 2.*EZSD)
      EYSCC = TAUY*(TAUY*(GY*EY - EYS) - 2.*EYSD)
      EZSSC = TAUZ*(EZSC/TAUL    + EZS - EZSS)
      EYSSC = TAUY*(EYSC/TAUL    + EYS - EYSS)
C
C**GRAVITY AND RATE BIAS
      WCC = EZSS + GBIAS
      WRC = EYSS + RBIAS
C
C
C**BODY RATE SHAPING AND GYRO DYNAMICS
      WQSC = WQSP
      WRSC = WRSP
      WQSCD = 94.2*(94.2*(WQ - WQS) - 2.*WQSD)
      WRSCC = 94.2*(94.2*(WR - WRS) - 2.*WRSD)
      IF (ABS(WQS) .LE. 30.) GO TO 30
      WQS = SIGN(30., WQS)
      VAR(NHSUM+3) = WQS
      IF (SIGN(1., WQSP) .NE. SIGN(1., WQS)) GO TO 30
      WQSP = 0.
      WQSC = 0.
      VAR(NHSUM+2) = WQSP
   30 IF (ABS(WRS) .LE. 30.) GO TO 32
      WRS = SIGN(30., WRS)
      VAR(NHSUM+5) = WRS
      IF (SIGN(1., WRSP) .NE. SIGN(1., WRS)) GO TO 32
      WRSP = 0.
      WRSC = 0.
      VAR(NHSUM+4) = WRSP
   32 CONTINUE
C
      IF(ABS(UQS).GT.30.)UQS=WQSD/80.+WQS
      IF(ABS(URS).GT.30.)URS=WRSD/80.+WRS
      UQSC=((WQSD/80.+WQS)-UQS)*800.
      URSC=((WRSD/80.+WRS)-URS)*800.
C
C**SUMMATION OF RATE DAMPING AND GUIDANCE SIGNALS AND THEIR DERIVATIVES
      EZRR = UQS - WQC
      EYRR = URS - WRC
C
      UKR = .95
      IF (T .LT. TCY2) UKR = 4.25
      IF (T .LT. TCY1) UKR = 0.
C
      ESUMDC = UKR*(EZRR - EYRR)
      ESUMEC = UKR*(EZRR + EYRR)
      IF(ABS(ESUMEC).GT.60.)ESUMED=SIGN(60.,ESUMED)
```

```
      IF(ABS(ESUMCC).GT.6C.)ESUMCC=SIGN(60.,ESUMOD)
C
C**TCTAL GUIDANCE SIGNAL SHAPING AND LIMITING
      E1CCC=1CCO.*((ESUMCC/8.+ESUMO)-E1DCR)
      EVNC1=1C00.*((ESUMEC/8.+ESUME)-EVNDR)
      IF(ABS(EVND1).GT.1CCO.)EVNCR=ESUME+ESUMED/8.
     1-SIGN(1.,EVNC1)
      IF(ABS(E1CCC).GT.1C00.)E1CCR=ESUMO+ESUMOD/8.
     1-SIGN(1.,E1CCC)
      IF(ABS(EVND1).GT.1000.)EVNC1=SIGN(1C00.,EVND1)
      IF(ABS(E1CCC).GT.1C00.)E1CCC=SIGN(1CCO.,E1DDC)
      EVNCC=EVNCR
      ECCCC=E1CCR
      IF (ABS(ECDCR) .GT. HLIMO) ECDCR = SIGN(HLIMO, EODCR)
      IF (ABS(EVNCR) .GT. HLIME) EVNCR = SIGN(HLIME, EVNCR)
C
C**RCLL SIGNAL SHAPING
      UKP = .33
      IF (T .LT. TCY2) UKP = 1.65
      IF (T .LT. TCY1) UKP = 0.
      UPHIS = UKP*(BPHIC/12. + BPHI)
      BPHISC=36.*(UPHIS-BPHIS)
      BCELPT=-(BPHISC/108.+BPHIS)
      BCELPC=1CCO.*(BCELPT-BCELPC)
      BC=BCMAX/10.
      IF(ABS(BPHIS).GT.BC)BPHIS=SIGN(BD,BPHIS)
      IF(ABS(BCELPC).GT.5CC.)BCELPC=BCELPT-SIGN(.5,BDELPO)
      IF(ABS(BCELPC).GT.5CC.)BCELPD=SIGN(5CO.,BDELPO)
      IF(ABS(BCELPC).GT.BC)BCELPC=SIGN(BD,BDELPC)
      IF((T.GT..9).AND.(T.LT..9C3))
     1WRITE(6,6)(I,C(I),C(I+1),C(I+2),C(I+3),C(I+4),C(I+5),C(I+6),I=1,35
     *10.7)
    6 FCRMAT(1H1/(I5,1P7E15.7))
C
C**AUTCPILCT CUTPUT CURRENTS TO EACH ACTUATUR (FROM SUMMATION AMPS)
      BCELTC(1) = EOCCR - BDELPC
      BCELTC(2) = EVNCR - BCELPC
      BCELTC(3) = EOCCR + BCELPC
      BCELTC(4) = EVNCR + BCELPC
      RETURN
      END
```

Distribution List
Not Filmed

Page 133 & 134

UNCLASSIFIED

Security Classification

DOCUMENT CONTROL DATA - R & D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY (Corporate author) Department of Mechanical Engineering University of Florida Gainesville, Florida	2a. REPORT SECURITY CLASSIFICATION Unclassified 2b. GROUP - - -

3. REPORT TITLE

CLOSE AIR SUPPORT MISSILE GUIDANCE AND CONTROL STUDY
Volume I. Six-Degree-of-Freedom Simulation

4. DESCRIPTIVE NOTES (Type of report and inclusive dates)
Final Report (9 December 1970 to 9 December 1971)

5. AUTHOR(S) (First name, middle initial, last name)

J. Mahig

6. REPORT DATE	7a. TOTAL NO. OF PAGES	7b. NO. OF REFS
December 1971	142	None

8a. CONTRACT OR GRANT NO. F08635-71-C-0073 b. PROJECT NO. 670B c. Task No. 01 d. Work Unit No. 010	9a. ORIGINATOR'S REPORT NUMBER(S) 9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report) AFATL-TR-71-169

10. DISTRIBUTION STATEMENT

T&E

Distribution limited to U. S. Government agencies only; ~~this report documents the close air support missile guidance and control study;~~ distribution limitation applied December 1971. Other requests for this document must be referred to the Air Force Armament Laboratory (DLWG), Eglin Air Force Base, Florida 32542.

11. SUPPLEMENTARY NOTES	12. SPONSORING MILITARY ACTIVITY
Available in DDC	Air Force Armament Laboratory Air Force Systems Command Eglin Air Force Base, Florida 32542

13. ABSTRACT

This report describes in detail a six-degree-of-freedom program which can be used to determine the trajectory and miss distance of a missile system. The options for the program are such as to permit variation of the aerodynamics, seeker, autopilot actuator and missile motor performance for the purpose of accurately simulating a given missile design and evaluating the effects of changes in system parameters. Sufficient detail has been included in the text in order to minimize the users' effort needed to know how to update or modify the program for his purposes.

DD FORM 1 NOV 65 1473

UNCLASSIFIED

Security Classification

14 KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
Guided Missile						
Missile Simulation System						
Six-Degree-of-Freedom Simulation						